OMEGA

Zdzisław Pawlak

MASZYNA I JĘZYK

PW

Doc. dr Zdzisław Pawlak urodził się dnia 10 listopada 1926 roku w Łodzi, gdzie ukończył szkołę średnią i w roku 1947 rozpoczął studia na Wydziale Elektrycznym Politechniki Łódzkiej. W roku 1949 przeniósł się na Wydział Łączności Politechniki Warszawskiej. Od ukończenia studiów w 1951 roku do chwili obecnej pracuje w Instytucie Matematycznym PAN pełniąc kolejno funkcje asystenta, adiunkta i docenta. Doc. dr Pawlak poświęcił się badaniu problemów organizacji maszyn matematycznych; na ten temat publikował liczne prace i artykuły w czasopismach krajowych i zagranicznych.

Ω

8

Współczesna Biblioteka Naukowa

Omega

Zdzisław Pawlak

Maszyna i język

Warszawa 1964
Państwowe Wydawnictwo Naukowe

Okładkę projektował: Tadeusz Pietrzyk

4H.SZ.until.E.of.T!

Państwowe Wydawnictwo Naukowe — Warszawa 1964

Redaktor: Ryszard Herczyński
Redaktor techniczny: Leokadia Lass

Wydanie I. Nakład 20.000 + 270 egz.
Ark. wyd. 3,75. Ark. druk. 7.
Papier ilustracyjny kl. V, 70 g
Oddano do składania 18 XI 1963.
Podpisano do druku w marcu 1964.
Druk ukończono w kwietniu 1964.
Cena w subskrypcji zł. 10,—

Zakł. Graf. im. M. Kasprzaka w Poznaniu 2122/63 — O-8

Spis treści

Wstęp 9

Część I Procesy

1. Obliczenie 19
   Porządek obliczenia 21
   Porządek poprzeczny 22
   Porządek wzdłużny 23
   Liniowe przedstawienie obliczenia 25
2. Opis obliczenia 31
   Język podstawowy 32
   Język uproszczony 35
3. Organizacja maszyn cyfrowych 37
   Maszyna realizująca język podstawowy 38
   Maszyna realizująca język uproszczony z porządkiem P 41
   Maszyna realizująca język uproszczony z porządkiem W 47
   Wykonywanie bardziej skomplikowanych obliczeń 49
4. Inne języki 52
   Języki nawiasowe 52
   Uproszczony język nawiasowy 59
   Język Łukasiewicza 61
5. Składanie przedmiotów 65
   Produkcja liniowa 66
   Skręcanie powrozów 71
   Budowa mozaiki. Składanie klocków 73

Część II Systemy

6. Systemy i hierarchie 79
   Struktura sznurka 80
   Struktura książki 82

Dorzecze 85
Inne systemy 89

7. Semantyczne i syntaktyczne określenie języka . . 91
Semantyka 91
Struktura formuł 93
Metajęzyk 98

8. Składnia języka potocznego 102

Zakończenie 107

Indeks pojęć 110

Bibliografia 112

„Powstanie dużych maszyn matematycznych, gwałtowne doskonalenie biegłości i pewności ich działania, to jedno z tych niespodziewanych historycznych zdarzeń, które rodzą trudne do przewidzenia konsekwencje”.

Hao Wang

Wstęp

Maszyny matematyczne są niewątpliwie jednym z najdziwniejszych wytworów człowieka. Pojawienie się ich wywołało i nadal wywołuje duże poruszenie w wielu dziedzinach nauki: matematyce, fizyce, chemii, biochemii, psychologii, neurologii, ekonomii, socjologii, językoznawstwie i wielu innych. Nie chodzi mi tu o praktyczne znaczenie maszyn matematycznych w tych dziedzinach, o to, że można za ich pomocą to czy owo bardzo szybko obliczyć. Rola maszyn matematycznych w tym przypadku jest powszechnie znana. Równie ważny, a może i ważniejszy jest drugi aspekt maszyn matematycznych. Okazało się, że problemy teoretyczne występujące w związku z ich konstrukcją mają bardzo ogólny charakter i mogą się odnosić nie tylko do techniki, ale i do wielu innych, nawet dość odległych dziedzin.

W związku ze studiami autora nad tzw. maszynami bezadresowymi okazało się, że szczególną rolę odgrywają tu pojęcia **procesu, systemu** i **języka**. Pojęcia te mają jednak charakter na tyle ogólny, że badanie ich własności można prowadzić nie tylko pod kątem zastosowania do maszyn matematycznych. Pojęcie procesu np. spotykamy niemal na każdym kroku. Mówimy o procesie uczenia się, starzenia, procesie produkcji, procesie obliczenia,

rozumowania itp. Co jest wspólnego w tych wszystkich pojęciach? Czy można je badać niezależnie od tego, czy dotyczą one np. procesu produkcyjnego, czy też procesu rozumowania formalnego? Czy można podać jakieś ogólne własności procesów, nie wchodząc w to, czego ten proces dotyczy?

Przez proces będziemy tutaj rozumieli ciąg czynności, które z pewnych obiektów tworzą nowe obiekty. A więc procesem w naszym rozumieniu nie będzie np. uczenie się, gdyż nie ma tu wyraźnie występującego momentu t w o r z e n i a z jednych obiektów innych obiektów. Natomiast procesem według tej definicji jest proces chemiczny, gdyż w procesie chemicznym powstają z jednych związków inne związki chemiczne. Podobnie procesem w myśl naszej definicji będzie proces produkcji, np. samochodu. Wtedy bowiem z jednych podzespołów — za pomocą określonych operacji — tworzymy nowe zespoły. Procesem będzie również obliczanie, gdyż z jednych napisów przedstawiających liczby, za pomocą operacji arytmetycznych: dodawania, odejmowania, mnożenia i dzielenia, otrzymamy nowe napisy, przedstawiające wynik obliczenia. Obiektami tutaj są napisy; operacje służą do przekształcania napisów. Także wnioskowanie formalne jest procesem, polega ono bowiem na przekształcaniu napisów-przesłanek w myśl określonych reguł logicznych, celem otrzymania napisów-wniosków.

Rozpatrywanie tak ogólnych procesów byłoby dość trudne i dlatego będziemy tutaj rozpatrywać tylko procesy, które nazwiemy prostymi. Przez proces prosty będziemy rozumieli tylko taki proces, w którym w wyniku każdej operacji powstaje jeden obiekt. Procesem prostym jest np. obliczenie, gdyż w wyniku każdej operacji powstaje jeden wynik. Również wnioskowanie jest procesem prostym, gdyż w wyniku każdej operacji wnioskowania powstaje z przesłanek jeden wniosek. Jed-

nakże działanie kwasem na tlenek metalu nie jest procesem prostym, gdyż w wyniku takiej operacji powstają dwa nowe związki: sól tego metalu i woda — więc dwa obiekty, co nie jest zgodne z naszą definicją. Strukturę procesu prostego można sobie wyobrazić w postaci odwróconego drzewa. Gałęzie reprezentują obiekty, których proces dotyczy; rozgałęzienia natomiast — operacje występujące w procesie. W wyniku operacji, z kilku gałęzi powstaje nowa gałąź; rezultatem końcowym procesu jest natomiast pień drzewa.

Można również rozróżnić proces analizy od procesu syntezy. W procesie syntezy składamy nowe przedmioty z przedmiotów już złożonych, w procesie analizy postępujemy odwrotnie: złożony obiekt rozkładamy na jego części składowe. Tak więc np. jeżeli procesem syntezy byłoby splatanie sznurka, to procesem jego analizy będzie rozplatanie sznurka na części składowe. W tej książce zajmiemy się tylko procesami syntezy.

Z pojęciem procesu związany jest problem realizacji. Operacje arytmetyczne w procesie liczenia mogą być wykonywane przez człowieka, ale można je również wykonywać za pomocą odpowiedniej maszyny — arytmometru. W procesie chemicznym operacje wykonywane są jeszcze na innej zasadzie bez dodatkowych czynności człowieka, a tylko w odpowiednich warunkach: temperatury, ciśnienia itp. Warunkiem, aby z dwu składników powstał nowy związek, może też być obecność innego związku chemicznego. Sprawą tą nie będziemy się tutaj zajmowali, tj. nie interesuje nas w jaki sposób są wykonywane operacje zachodzące w procesie. Natomiast interesująca dla nas jest o r g a n i z a c j a p r o c e s u, tzn. w jaki sposób, w jakiej kolejności muszą być wykonywane operacje i jaka jest ogólna s t r u k t u r a urządzenia realizującego zadany proces. W przypadku procesu rachunkowego interesuje nas, jaka

powinna być struktura m a s z y n y matematycznej, która by dany proces realizowała, oraz badanie różnych maszyn tego rodzaju. W przypadku procesu rozumowania, pytamy o strukturę maszyny służącej do wykonywania rozumowań; w przypadku procesu produkcyjnego — o organizację maszyny, czy też całej fabryki, realizującej dany proces. Jakie taka fabryka musi posiadać oddziały i jak one muszą wzajemnie współpracować, aby fabryka realizowała zadany proces produkcyjny.

W tym ostatnim przypadku całą fabrykę traktujemy jako maszynę. Jak z tego widać pojęcie m a s z y n y, którego tu używamy jest bardzo szerokie i nie zawsze pokrywa się z potocznym rozumieniem tego słowa. Maszyną dla nas jest każdy układ realizujący odpowiedni proces, przy czym ograniczamy się do maszyn realizujących proces prosty, tj. maszyn, które wytwarzają tylko jeden obiekt. Wracając do naszego przykładu fabryki, jest to fabryka, której asortyment wytworów ograniczony jest do jednego obiektu.

Badając maszynę interesuje nas, czy i jak jej organizacja jest związana z procesem, który ma ona realizować. Chcemy odpowiedzieć na pytanie, czy struktura procesu wyznacza np. strukturę fabryki. Okazuje się, że takie powiązanie w istocie ma miejsce i co więcej, że w dość podobny sposób możemy badać zarówno strukturę maszyn matematycznych, jak i schematy organizacyjne fabryk.

Aby dowolny proces realizować, potrzebny jest opis przebiegu tego procesu, tzw. program. Do opisu takiego potrzebny jest j ę z y k. Należy zatem sprecyzować pojęcie języka, w którym dany proces jest opisywany. Na ogół języków takich można podać wiele i interesujące jest badanie ich własności.

W pierwszej części książki, zatytułowanej *Procesy*, ograniczono się w zasadzie do procesów ob-

liczania i produkcji, wykazując, że oba te procesy można w dużym stopniu traktować identycznie, tzn. schematy maszyn matematycznych, służących do realizowania procesów obliczeniowych można traktować jako schematy fabryki, realizującej zadany proces technologiczny i odwrotnie. Warto dodać, że obecnie istniejące maszyny matematyczne działają na innej zasadzie niż te, które zostały podane tutaj. Można jednak przypuszczać, że przedstawione koncepcje znajdą w przyszłości zastosowanie praktyczne.

Celem naszym zresztą nie jest zapoznanie Czytelnika z działaniem budowanych obecnie maszyn matematycznych, lecz pokazanie różnych aspektów języka. Maszyny matematyczne są tutaj jedną z możliwych interpretacji.

Druga część książki poświęcona jest *systemom*. Co to jest system? Pewne obiekty tworzące razem jakąś całość nazywamy **systemem**. Mówimy np. system słoneczny, mając na myśli Słońce wraz z powiązanymi z nim planetami. System energetyczny — to powiązany, współpracujący ze sobą zbiór elektrowni. System zarządzania stanowi zespół instytucji, wykonujących wspólnie określone zadanie administracyjne. Zespół idei stanowiący jakąś całość, nazywany jest systemem filozoficznym. Systemem jest zespół elementów dowolnej maszyny. Organizm biologiczny jest systemem. Systemem jest również społeczeństwo. Z pojęciem systemu, podobnie jak z pojęciem procesu, spotykamy się na każdym kroku.

Nic więc dziwnego, że czynione są próby stworzenia ogólnej teorii systemów, która by obejmowała dowolne systemy i pozwalała na badanie ich własności niezależnie od tego, z jakich elementów systemy te się składają. Próby tego rodzaju są dopiero w stadium początkowym i trudno przewidzieć ich wynik. Nie jest naszym zadaniem przedstawienie obecnego stanu teorii systemów,

ale podanie pewnych związków między pojęciem systemu i języka. Będziemy się, mianowicie, zajmowali zagadnieniem, w jaki sposób można strukturę systemu przedstawić za pomocą języka i jak struktura języka zależy od systemu, który ma on przedstawiać.

Pojęcie systemu wiąże się ściśle z pojęciem procesu. Z jednej strony w wyniku jakiegoś procesu powstaje obiekt złożony, stanowiący system, np. samochód. Z drugiej strony fabryka, w której dany proces — w naszym przypadku proces produkcji samochodu — zachodzi, jest również zespołem wzajemnie powiązanych części, a więc jest również systemem. W pewnym systemie zachodzi więc proces realizujący nowy system. Szczególnie interesujące są tutaj przykłady z biologii. Nasienie rośliny możemy uważać za system. Z systemu tego tworzy się roślina, która produkuje nowe ziarno, nowy system podobny do tego, z jakiego powstała. A więc tak, jak gdyby fabryka produkowała nową fabrykę, ta zaś następną i tak dalej.

Mówiąc dotąd o języku, mieliśmy na myśli języki sztuczne, formalne. Języki te są w wielu wypadkach dużo dogodniejsze od języka naturalnego, w którym sens poszczególnych terminów jest nieraz wieloznaczny i nie dość sprecyzowany. Najbardziej jaskrawym i powszechnie znanym (przynajmniej w elementarnym zakresie) przykładem języka sztucznego jest język matematyki.

Nasuwa się pytanie, jaki jest związek języków sztucznych z językami „naturalnymi" — polskim, rosyjskim itd. Wiadomo, że języki naturalne powstały i kształtowały się pod wpływem najrozmaitszych potrzeb i czynników i dlatego formalne ujęcie ich własności jest bardzo trudne, a ich pełny formalny opis — wręcz niemożliwy. Tym niemniej próby zastosowania aparatu matematycznego do badania języków naturalnych zostały od niedawna podjęte. Powstała nowa wiedza — lingwi-

styka matematyczna. W książce tej chcemy zwrócić uwagę na nieco inny, lecz nie mniej ważny sposób podejścia do języka naturalnego. Składnia języka naturalnego jest przystosowana raczej do wyrażania stosunków między przedmiotami aniżeli do opisywania czynności. Być może, człowiek widział kiedyś świat jako zbiór niepowiązanych ze sobą obiektów, zaś mowa mogła służyć wtedy tylko do nazywania przedmiotów. Dopiero pojawienie się konieczności opisywania stosunków, zachodzących między przedmiotami otaczającego świata, mogło być zaczątkiem powstania języka.

Oczywiście, o dalszym rozwoju języka decydował szereg innych, trudnych dziś do odtworzenia czynników. Tym niemniej zasadniczy szkielet składni wydaje się wskazywać, że jest on raczej językiem systemu niż językiem procesu. Dlatego też w części książki poświęconej systemowi umieściliśmy tzw. syntaktyczną analizę zdań w języku naturalnym, opracowaną ok. 30 lat temu przez zmarłego niedawno prof. K. Ajdukiewicza. Metoda ta została w r. 1950 rozszerzona przez Bar-Hilella, profesora logiki uniwersytetu w Jerozolimie, i stanowiła podstawy pierwszego mechanicznego tłumaczenia, dokonanego za pomocą maszyny matematycznej w Stanach Zjednoczonych.

Przedstawiony powyżej krąg zagadnień, jak widać choćby z przytoczonego ich opisu, nie mieści się w ramach jednej dyscypliny naukowej. Znajduje się on na styku pewnych gałęzi matematyki (logiki matematycznej i teorii struktur) teorii organizacji, techniki i językoznawstwa. Mówiąc ogólnie, należy on do zakresu zainteresowań cybernetyki, lecz czytelnik, który już się z jakimś jej wykładem spotkał, zauważy, że proponowane tu podejście jest odmienne od powszechnie przyjętego.

W związku z powyższym chciałbym powiedzieć, że w niniejszej książce unikałem modnych obecnie

uogólnień i spekulacji, które często wiąże się z cybernetyką a nawet włącza w zakres jej rozważań. Starałem się w sposób dostępny pokazać te problemy i trudności, które pojawiają się już w związku z teorią stosunkowo prostych maszyn, w szczególności z teorią maszyn cyfrowych. Oczywiście, bez ich rozwiązania trudno mówić o zrozumieniu układów bardziej skomplikowanych, w szczególności zaś układów biologicznych.

Jest rzeczą wielce prawdopodobną, że „język" tych układów jest odmienny niż ten, który rozważamy w tej książeczce. Dotyczy to zarówno tzw. kodu genetycznego, za pomocą którego przekazywana jest informacja biologiczna z pokolenia na pokolenie, jak i w jeszcze większym stopniu „języka", w którym pracuje mózg. O tym języku prawie nic, jak dotąd, nie wiadomo. Jest rzeczą niewątpliwą, że „język" ten jest blisko związany z istotą pracy mózgu. Ekstrapolowanie naszej, dość skąpej nawet w zakresie maszyn matematycznych, wiedzy o ich działaniu na obiekt tak skomplikowany jest zupełnie nieusprawiedliwione.

Sądzę, że badanie takich pojęć jak proces, system i język pozwoli wyjaśnić wiele spraw nie tylko w teorii maszyn matematycznych, lecz także w ekonomii, biologii i językoznawstwie. Jednak by badania te naprawdę okazały się płodne, trzeba zacząć od spraw podstawowych. Takim podstawowym problemom poświęcona jest właśnie niniejsza książka.

Na zakończenie chciałbym wyrazić podziękowanie dr A. Ehrenfeuchtowi oraz dr R. Herczyńskiemu za szereg cennych uwag. Dr Herczyńskiemu jestem ponadto winien szczególną wdzięczność za dużą pomoc w zredagowaniu tej książki.

Część I

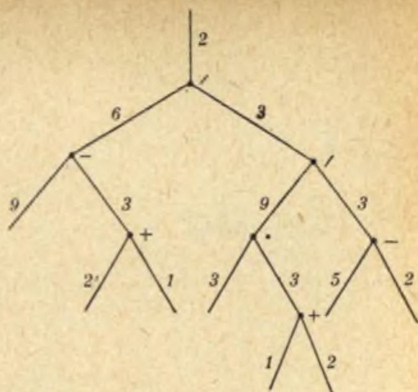
Procesy

1

Obliczenie

Wiadomo, co to jest obliczenie, lub, jeśli kto woli, rachunek. Mamy zadane jakieś liczby, oraz określone działania na tych liczbach, np. dodawanie, odejmowanie, mnożenie i dzielenie. Wykonując naznaczone działania na danych początkowych otrzymujemy nowe liczby — wyniki częściowe. Na tych wynikach częściowych wykonujemy dalsze działania, otrzymując dalsze wyniki częściowe itd. ..., aż do otrzymania wyniku końcowego.

Rozpatrzmy nieco szczegółowiej proces rachowania. Dla ułatwienia będziemy przedstawiali rachunek w postaci rysunku, jak to pokazano niżej (rys. 1). Działania dodawania, odejmowania, mnożenia i dzielenia oznaczamy odpowiednio +, −, · /. Rysunek taki przypomina odwrócone drzewo. Kropki (rozgałęzienia) oznaczają działania, kreski (gałęzie) oznaczają liczby. Każde rozgałęzienie łączy trzy gałęzie: dwie od dołu i jedną od góry. Obie gałęzie dolne przedstawiają liczby, na których jest wykonywane działanie; gałąź górna reprezentuje liczbę, która jest wynikiem działania. Liczby, na których jest wykonywane działanie nazywamy jego **argumentami**. W dalszym ciągu będę



Ryc. 1

używał tej nazwy. Najwyżej położona gałąź oznacza wynik końcowy obliczenia. Gałęzie bez kropek u dołu nie są wynikami częściowymi, przedstawiają **dane,** lub jak mówią czasem, dane początkowe. W podanym przykładzie dla uproszczenia przyjęto liczby jednocyfrowe, co oczywiście nie ma znaczenia. Równie dobrze można by przyjąć liczby o dowolnej ilości cyfr.

Warto zwrócić uwagę, że drzewo jest podzielone jak gdyby na piętra. Najwyżej znajduje się jedno (ostatnie) działanie. Piętro niżej narysowane są dwa działania na jednym poziomie, jeszcze niżej — trzy działania na jednym poziomie i najniżej „na parterze" — jedno działanie. Podany sposób przedstawienia rachunku pozwala na łatwe uchwycenie struktury całego obliczenia. W podobny sposób, jak się przekonamy, można przedstawić inne rodzaje obliczeń.

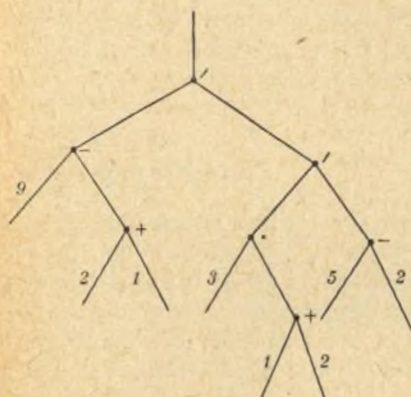
Rys. 1 przedstawia obliczenie po wykonaniu. Jeżeli chcemy przedstawić obliczenie przed wykonaniem, wpisujemy do drzewa tylko wszystkie działania oraz dane początkowe bez wyników częściowych. Podobnie można przedstawić obliczenie

w dowolnym stadium wykonywania, wpisując tylko te wyniki częściowe, które zostały już obliczone.

Porządek obliczenia

Na rys. 2 przedstawione jest obliczenie przed wykonaniem. Przed przystąpieniem do liczenia należy się zdecydować na jakąś kolejność wykonywania działań. Kolejność tę nazwiemy porządkiem obliczenia. Kolejność wykonywania działań w obliczeniu jest w dużym stopniu dowolna, byleby tylko nie zacząć od działania, którego argumenty nie są jeszcze obliczone. Możemy np. wykonywać obliczenie przedstawione na rys. 2 w jednym z następujących porządków:

$1 + 2 = 3$	$1 + 2 = 3$	$2 + 1 = 3$
$3 \cdot 3 = 9$	$2 + 1 = 3$	$1 + 2 = 3$
$5 - 2 = 3$	$3 \cdot 3 = 9$	$9 - 3 = 6$
$2 + 1 = 3$	$9 - 3 = 6$	$5 - 2 = 3$
$9 / 3 = 3$	$5 - 2 = 3$	$3 \cdot 3 = 9$
$9 - 3 = 6$	$9 - 3 = 6$	$9 / 3 = 3$
$6 / 3 = 2$	$6 / 3 = 2$	$6 / 3 = 2$



Ryc. 2

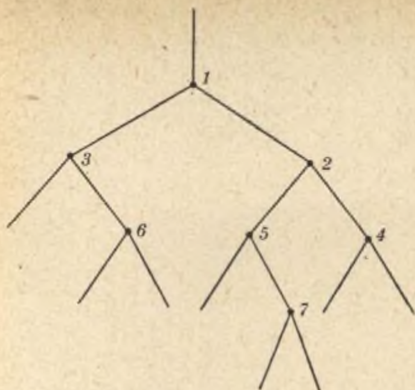
Milcząco założyliśmy, że w danej chwili wykonywane jest tylko jedno działanie. Możemy sobie z łatwością wyobrazić, że jedno obliczenie wykonuje np. dwóch rachmistrzów. Pierwszy zaczyna od lewej strony drzewa, drugi — od prawej. Rachmistrzów może być więcej. Pojawia się wtedy szereg kłopotliwych problemów, jak zorganizować właściwie wszystkim rachmistrzom pracę, aby sobie wzajemnie nie przeszkadzali i byli jak najlepiej wykorzystani, tzn. aby żaden z nich nie był bez zajęcia. Zagadnieniem tym nie będziemy się zajmować.

W dalszym ciągu książki przyjmiemy, że w każdym obliczeniu wykonywane jest w danej chwili tylko jedno działanie. Obliczenia takie nazwiemy **sekwencyjnymi.** A więc będziemy się zajmowali tylko obliczeniami sekwencyjnymi.

Ograniczymy się również do rozpatrywania tylko dwóch porządków obliczenia, które dla maszyn matematycznych mają zasadnicze znaczenie. W obu tych porządkach działania nie są wykonywane w dowolnej kolejności, ale w myśl pewnych, łatwych do uchwycenia reguł. Pierwszy porządek obliczania nazwiemy poprzecznym, drugi — wzdłużnym. (W dalszych rozważaniach pierwszy będzie oznaczać *P*, drugi — *W*). Oba porządki obliczenia omówione są dalej.

Porządek poprzeczny

Porządek poprzeczny charakteryzuje się tym, że działania wykonywane są piętrami, poczynając od piętra najniższego, w kierunku od lewej do prawej strony. Inaczej mówiąc każde działanie znajdujące się na wyższym piętrze jest wykonane później od każdego działania z piętra niższego; jeżeli działania znajdują się na tym samym piętrze, to działanie znajdujące się na **prawo** jest wykonywa-



Ryc. 3

ne później od działania, znajdującego się na lewej stronie piętra.

Ewentualne wątpliwości wyjaśnia przykład na rys. 3. Dla ułatwienia, działania są ponumerowane liczbami od 1 do 7 i wykonywane są w kolejności numerów malejących, tj. 7, 6, 5, 4, 3, 2, 1.

Przebieg obliczenia pokazanego na rys. 1 w porządku P jest następujący:

7. $1 + 2 = 3$
6. $2 + 1 = 3$
5. $3 \cdot 3 = 9$
4. $5 - 2 = 3$
3. $9 - 3 = 6$
2. $9 / 3 = 3$
1. $6 / 3 = 2$

Liczby z lewej strony są numerami działań (na rys. 3 nie podano działań ani liczb, które są takie same jak na rys. 2).

Porządek wzdłużny

Zasada numerowania działań w porządku wzdłużnym pokazana jest na rys. 4. Jak poprzednio,

Liniowe przedstawienie obliczenia

Do tej pory przedstawialiśmy obliczenia w formie rysunku. Sposób ten ma pewne zalety, przede wszystkim pozwala na ogarnięcie jednym spojrzeniem całego procesu obliczania, oczywiście o ile proces ten nie jest zbyt długi. Drugą zaletą jest łatwość ustalenia kolejności działań, według jednego z dwóch przyjętych schematów, *P* lub *W*.

Mając na uwadze rachunek mechaniczny, znacznie wygodniej i ekonomiczniej jest ustawić działania i argumenty w ten sposób, aby znajdowały się na jednej linii, w kolejności ich wykonywania poczynając np. od strony lewej ku prawej tak, jak czytamy każdy tekst pisany. Przyjmiemy tu następujący schemat **liniowego przedstawienia obliczenia**

$$a,b,A,c,d,B,...u,v,X,z$$

Duże litery oznaczają tu działania, natomiast małe litery stojące przed symbolem działania oznaczają jego argumenty.

Przykład podany na rys. 1 będzie miał teraz postać

1,2, +, 2,1, +, 3,3,., 5,2, −, 9,3, −, 9,3, / , 6,3,/,2 (*P*)

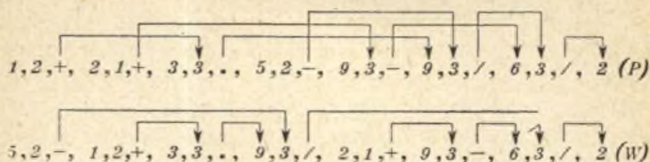
lub

5,2, −, 1,2, +, 3,3,., 9,3, / , 2,1, +, 9,3, −, 6,3,/,2 (*W*)

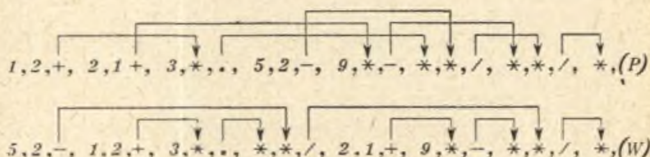
Litery w nawiasach oznaczają przyjęty porządek obliczenia. Warto podkreślić, że symbol działania pisany jest **po** odpowiadających mu argumentach. Można oczywiście pisać symbol działania między argumentami, tak jak to czyni się w szkole, lub przed nimi. Ze względów, o których powiemy niżej, wybraliśmy jednak odmienny zapis.

Niestety, w takim przedstawieniu obliczenia straciliśmy informację, które liczby są danymi początkowymi, a które wynikami częściowymi. Informację tę możemy dodać rysując na podstawie

drzewa strzałki, wskazujące wyniki częściowe odpowiednich działań w następujący sposób:



Tak wygląda liniowe przedstawienie obliczenia po jego wykonaniu. Podobnie możemy przedstawić obliczenie przed wykonaniem. Analizując rys. 2 wypisujemy kolejno działania wraz z argumentami w ten sposób, że jeżeli jakiś argument jest wynikiem częściowym, piszemy na jego miejsce np. umownie gwiazdkę * . Zależnie od przyjętego porządku obliczenia otrzymamy dwa wyrażenia:



Strzałki mają takie same znaczenie, jakie miały poprzednio. Zgodnie z przyjętą umową obliczenie należy wykonywać od strony lewej ku prawej, po jednym działaniu arytmetycznym, umieszczając wynik każdego działania na miejscu gwiazdki wskazanej przez strzałkę. Wykonanie pojedynczego działania będziemy nazywali **krokiem obliczenia.** Dla większej przejrzystości argumenty oraz symbole działań będziemy pisać w kratkach. Zbędne jest wtedy pisanie gwiazdki zamiast wyniku częściowego, gdyż jest on oznaczony przez pustą kratkę. Poniżej przedstawiono kolejne kroki obliczenia dla porządków *P* i *W*. W odpowiednich kratkach nie wpisane są jeszcze wyniki częściowe przedstawionego kroku.

Porządek *P*

1	2	+	2	1	+	3		.	5	2	−	9		−			/			/		7
1	2	+	2	1	+	3	3	.	5	2	−	9		−			/			/		6
1	2	+	2	1	+	3	3	.	5	2	−	9	3	−			/			/		5
1	2	+	2	1	+	3	3	.	5	2	−	9	3	−	9		/			/		4
1	2	+	2	1	+	3	3	.	5	2	−	9	3	−	9	3	/			/		3
1	2	+	2	1	+	3	3	.	5	2	−	9	3	−	9	3	/	6		/		2
1	2	+	2	1	+	3	3	.	5	2	−	9	3	−	9	3	/	6	3	/		1

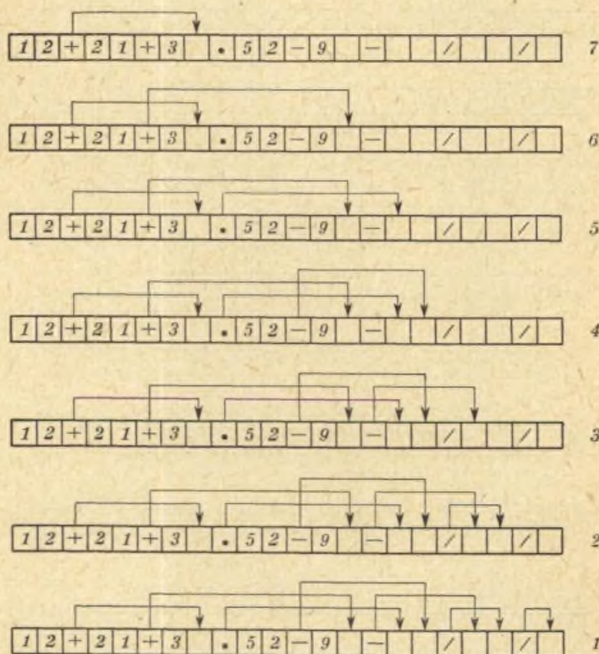
Porządek *W*

5	2	−	1	2	+	3		.			/	2	1	+	9		−			/		7
5	2	−	1	2	+	3		.		3	/	2	1	+	9		−			/		6
5	2	−	1	2	+	3	3	.		3	/	2	1	+	9		−			/		5
5	2	−	1	2	+	3	3	.	9	3	/	2	1	+	9		−			/		4
5	2	−	1	2	+	3	3	.	9	3	/	2	1	+	9		−		3	/		3
5	2	−	1	2	+	3	3	.	9	3	/	2	1	+	9	3	−		3	/		2
5	2	−	1	2	+	3	3	.	9	3	/	2	1	+	9	3	−	6	3	/		1

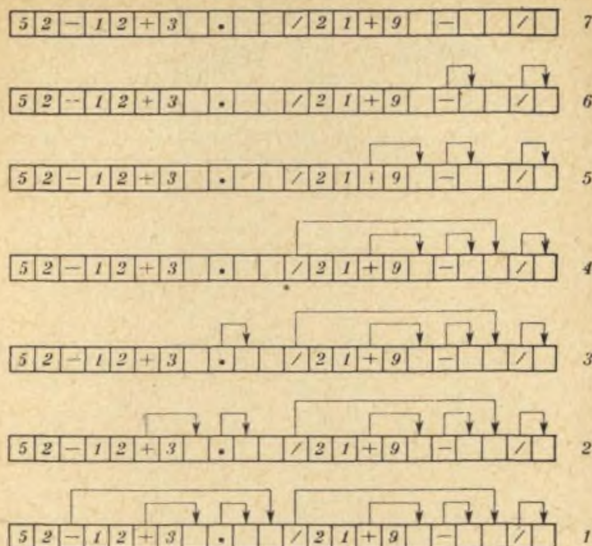
Analizując dokładniej przebieg obliczenia łatwo zauważyć zasadę umieszczania wyników częściowych. Dla porządku *P* wynik każdego kroku umieszczany jest w najbliższym na prawo pustym kwadracie. Identyczna zasada jest słuszna dla porządku *W* z tym, że przy pisaniu strzałek należy rozpatrywać działania od strony prawej do lewej, a nie jak poprzednio, od lewej do prawej.

Sposób wpisywania strzałek wskazujących, gdzie ma być wpisany wynik częściowy, pokazano niżej:

Porządek *P*



Podane zasady pisania strzałek są ogólne, tj. dla każdego obliczenia przedstawionego liniowo, jeżeli znamy kolejność wykonywania działań, po-



trafimy określić już teraz bez pomocy drzewa, gdzie należy umieścić wynik częściowy każdego kroku obliczenia.

W dalszym ciągu książki „obliczenie" będzie zawsze oznaczało obliczenie sekwencyjne, liniowe, wykonywane w porządku *P* lub *W*. Innych obliczeń nie będziemy rozpatrywali, chyba, że to będzie wyraźnie powiedziane.

Do tej pory używaliśmy wymiennie słów „obliczenie" i „rachunek". W dalszym ciągu przyjmiemy, że obliczenie to konkretny proces tworzenia nowych liczb z liczb już znanych, natomiast zbiór wszystkich takich procesów nazwiemy rachunkiem. Oczywiście, rachunek to nie tylko działania liczbowe. Jeżeli gałęzie drzewa będziemy rozpatrywali jako inne obiekty mate-

matyczne, np. jako wektory lub macierze, a punkty — jako działania wektorowe lub macierzowe, to otrzymamy rachunek wektorowy lub macierzowy. Można także, jak to pokażemy później, rozpatrywać gałęzie i punkty jako jeszcze inne obiekty. Na razie jednak, w najbliższych dwóch paragrafach zajmiemy się rachunkiem liczbowym.

2

Opis obliczenia

W bardzo dużych obliczeniach — jak wykazała praktyka — występuje nawet 10^{11} działań arytmetycznych. Gdybyśmy chcieli przedstawić takie obliczenie w dotychczasowej postaci trzeba by użyć $3 \cdot 10^{11}$ znaków (dwa znaki na argumenty i jeden na działanie). Zakładając, że na stronie maszynopisu mieści się 3000 znaków pisarskich, otrzymamy, że do przedstawienia omawianego procesu potrzeba 10^8 stron. Taka ilość stron tworzyłaby tom grubości 10 km. Wyłania się więc sprawa krótkiego zapisu długich obliczeń. Niestety w ogólnym przypadku niewiele tu można zrobić. Jeżeli mamy np. dodać do siebie milion różnych liczb, to liczby te muszą być gdzieś zapisane. Na szczęście, w praktyce sprawa nie wygląda tak groźnie. Problemy, które prowadzą do długich obliczeń dają się często sformułować na paru kartkach papieru. W naszych rozważaniach przyjmiemy, że obliczenia zawierają niewielką ilość działań, np. kilkanaście. Pozwoli to nam na ominięcie trudności, związanych z opisem obliczeń bardzo długich. A zatem tematem naszych rozważań będą niewielkie obliczenia.

Zaczniemy od zwrócenia uwagi na różnice mię-

dzy obliczeniem a jego opisem. Stwierdziliśmy, że obliczenie to proces, a więc zjawisko odbywające się w czasie. W kolejnych etapach procesu wykonywane są różne czynności, w wyniku których z danych obiektów tworzymy nowe obiekty.

Uczyliśmy się w szkole, że liczby to twory abstrakcyjne, a napisy 1, 2, 3 itd. to nazwy tych tworów. Tutaj przyjmiemy, że liczby to tyle co odpowiednie napisy na papierze, bądź sygnały elektryczne w maszynie liczącej. Obliczenie natomiast, to przetwarzanie jednych napisów w inne za pomocą ustalonych reguł, bądź też określone przetwarzanie sygnałów elektrycznych.

Mówiąc więc ściśle, na rys. 1 i 2 nie są przedstawione obliczenia, a opisy obliczeń. Ponieważ obliczenie to czynność przekształcania napisów, a opis tej czynności jest również napisem, może być to powodem pewnych nieporozumień. Np. proces obliczenia sumy tysiąca liczb zawiera tysiąc operacji dodawania, gdyż tysiąc razy trzeba wykonać dodawanie, aby otrzymać sumę końcową. Natomiast w opisie dodawania: „dodaj tysiąc liczb" operacja dodawania jest wymieniona jednokrotnie. Zakładamy, że wiemy o jakie liczby chodzi w dodawaniu i w opisie ich nie wymieniamy.

W dalszym ciągu podamy kilka języków, służących do opisywania obliczeń, mając na uwadze krótkość zapisu oraz łatwość posługiwania się językiem. Będziemy także brali pod uwagę jego przydatność dla zastosowania w maszynach liczących. Tą ostatnią sprawą zajmiemy się dokładniej dopiero w następnym paragrafie.

Język podstawowy

Niektóre cechy obliczenia nie zależą od liczb na jakich jest ono wykonywane. Np. kolejność wykonywania działań, sposób umieszczania wyników

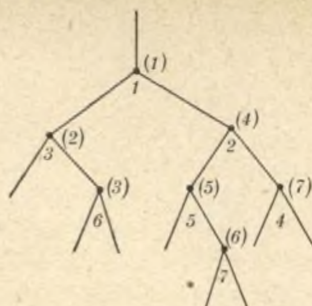
częściowych i inne. Tak np., gdy w drzewie przedstawionym na rysunku 1 wstawimy na miejsce umieszczonych tam działań zupełnie inne, to ogólna struktura obliczenia pozostanie bez zmiany. Ponieważ interesujemy się tu nie poszczególnymi obliczeniami a ich ogólną strukturą, to wygodnie nam będzie zamiast liczb pisać małe litery, zamiast działań — duże litery. Przechodzimy tym samym jak gdyby do wyższej klasy, w której uczą już nie arytmetyki lecz algebry. Małe litery oznaczają więc pewne obiekty (liczby), duże litery — czynności (działania na liczbach). Używając terminów gramatycznych małe litery można nazwać rzeczownikami, duże — czasownikami.

Napis

$$a,b\ A\ c,d\ B\ \dots\ u,v\ X\ z$$

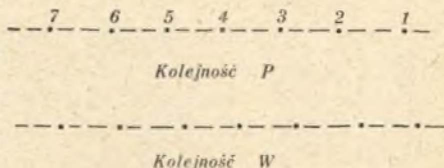
nazwiemy **formułą, wzorem, schematem, programem** albo **algorytmem obliczenia.** W terminologii gramatycznej byłoby to zdanie. Formuła nie określa jednoznacznie obliczenia, gdyż nie wiemy, które argumenty są danymi, a które wynikami częściowymi. Musimy więc odróżniać dane początkowe od wyników częściowych, ponadto musimy wiedzieć, wynikiem którego działania jest dany wynik częściowy. Jak wiemy z poprzednich przykładów, przy kolejności *P* lub *W* nie potrzebujemy oznaczać wyników częściowych różnymi literami, wystarcza do tego jeden symbol, gdyż związek między działaniami i odpowiadającymi im wynikami częściowymi zależy tylko od ich wzajemnego rozmieszczenia w formule.

Tak więc, obliczenie scharakteryzowane jest formułą postaci podanej wyżej (z tym, że wszystkie wyniki częściowe oznaczone są symbolem $_*$) oraz podaną kolejnością obliczenia: *P* lub *W*. Mówiąc krócej, formuła to odpowiednio spłaszczone



Liczby bez nawiasów podają kolejność P

Liczby w nawiasach podają kolejność W



Ryc. 5

drzewo. Jeżeli wszystkie punkty drzewa oznaczymy dużymi literami, gałęzie końcowe — małymi literami, pozostałe gałęzie — symbolem *, oraz wypiszemy wszystkie elementy drzewa w kolejności *P* lub *W* według schematu: lewa gałąź dolna, prawa gałąź dolna, rozgałęzienie, to otrzymamy formułę zapisaną w języku podstawowym. **Formuła jest więc liniowym obrazem struktury drzewa.**

Zbiór wszystkich formuł tworzy **język.** Obliczeniu odpowiada formuła, rachunkowi — język.

Dlaczego użyto tu nazwy: język, wyjaśnia przykład. Wyobraźmy sobie następującą zabawę. Są

dwie osoby, które mogą się porozumiewać ustnie, ale wzajemnie się nie widzą. Każda z osób posiada kartkę i ołówek. Jedna z nich rysuje dowolne drzewo według reguł, które omawialiśmy w drugim paragrafie. Po narysowaniu zadaniem jej jest opowiedzenie narysowanego obrazu w ten sposób, aby druga osoba mogła narysować identyczny obraz. Wystarczy, aby rysująca osoba mówiła w ustalonej uprzednio kolejności (*P* lub *W*) trzy słowa: daną początkową, wynik częściowy, działanie, a więc żeby wymawiała ona formułę danego drzewa. Przyjęliśmy tu, że nie interesują nas jakie litery są przyporządkowane rozgałęzieniom, oraz jakie litery odpowiadają gałęziom. Oczywiście, uwzględnienie oznaczeń literowych nie przedstawia trudności. Nawiązując do alfabetu Morse'a rysunki drzew możnaby przekazywać za pomocą trzech znaków: kropka — (działanie), krótka kreska — (dana), długa kreska — (wynik częściowy, wynik). Patrz rys. 5.

Język uproszczony

W języku podstawowym używaliśmy dwu rodzajów formuł; w jednych występowały liczby, w drugich — litery. Formuły z liczbami opisywały jeden konkretny proces obliczania, formuły z literami nie dotyczyły konkretnego procesu a całej grupy procesów podobnych, o jednakowych działaniach, wykonywanych w jednakowej kolejności, różniących się tylko danymi początkowymi i zatem wynikami częściowymi. Choć te dwa rodzaje formuł są pojęciowo różne, nie będziemy jednak chwilowo robili między nimi różnicy. Do spraw tych wrócimy przy omawianiu organizacji maszyn matematycznych. W języku podstawowym, dane w formule oznaczaliśmy różnymi literami. Jeżeli w dowolnej formule wszystkie dane oznaczymy jednakowym symbolem,

np. Δ, to formułę tę nazwiemy formułą uproszczoną, a zbiór wszystkich formuł uproszczonych — językiem uproszczonym.

Obliczenie przedstawione na rys. 1 w języku uproszczonym będzie miało postać:

$$\Delta\Delta + \Delta\Delta + \Delta * \cdot \Delta\Delta - \Delta * - **/**/* \qquad (P)$$

$$\Delta\Delta - \Delta\Delta + \Delta * \cdot **/\Delta\Delta + \Delta * - **/* \qquad (W)$$

3

Organizacja maszyn cyfrowych

Wiemy już, co to jest obliczenie i co jest językiem obliczenia. Przystępujemy z kolei do omawiania maszyn, służących do wykonywania obliczeń.

Istnieją dwie zupełnie odmienne koncepcje maszyn cyfrowych. Pierwsza, jako punkt wyjściowy przyjmuje obliczenia; punktem wyjściowym drugiej jest język. Maszyny pierwszego rodzaju nazywane są **modelami.** Model składa się z urządzeń wykonujących elementarne operacje arytmetyczne, połączonych ze sobą w ten sposób, jak wskazuje drzewo obliczenia, tj. punkty drzewa są interpretowane jako urządzenia liczące, a odcinki — jako połączenia między nimi. Drzewo jest więc jednocześnie schematem maszyny. Modele, choć nie w tak prymitywnej postaci, o jakiej była mowa, znalazły pewne zastosowania, jednakże decydującą rolę odgrywają obecnie maszyny grupy drugiej, zwane **maszynami programowanymi** albo **algorytmicznymi.** One będą przedmiotem naszych dalszych rozważań.

Wróćmy do języka. Stworzyliśmy go po to, aby opisywać obliczenia, przebiegi pewnych zdarzeń. Otóż okazuje się, że opis obliczenia można interpretować również jako opis konstrukcji maszyny,

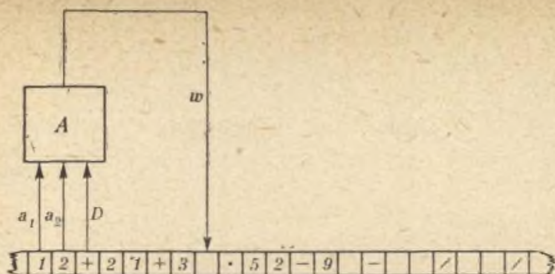
realizującej opisane obliczenie. Język, który zbudowaliśmy spełnia więc jednocześnie dwie role. W tym paragrafie zapoznamy się z drugą rolą języka. Podamy tylko kilka przykładów rozwiązań dla zilustrowania zasady technicznego interpretowania języka.

Podanie dobrych przykładów jest dość kłopotliwe, nie chcemy bowiem sięgać do spraw elektronowych, a bez tego niektóre rozwiązania mogą się wydawać nieuzasadnione. Wyłania się problem, który zasygnalizowaliśmy we wstępie. Proces zapisu i odczytu znaków na papierze różni się od analogicznego procesu w urządzeniu elektronowym, od zapisu i odczytu sygnałów elektrycznych w maszynie. Jeśli nie uwzględnimy tych różnic, to zaproponowane języki okażą się bądź nieprzydatne, bądź też zbyt trudne w praktycznej realizacji. Rozważane poniżej języki dobrano, mając na uwadze nie ich praktyczną przydatność, lecz prostotę. Ich zastosowanie w praktyce wymaga pewnych modyfikacji.

Maszyna realizująca język podstawowy

Idea maszyny programowanej polega na naśladowaniu liczenia za pomocą papieru i ołówka. Oczywiście nie jest to naśladowanie w znaczeniu dosłownym.

W maszynie programowanej mamy więc odpowiednik papieru, zwany **pamięcią maszyny.** Nazwa ta nie jest najlepsza, gdyż kojarzy się z nią pamięć w rozumieniu psychologów, jednakże oba te rodzaje pamięci nie mają wiele wspólnego. W różnych krajach dla uniknięcia niewłaściwych skojarzeń używa się innych nazw, np. magazyn (ang: *storage*). Pamięć maszyny służy do przechowywania formuły w trakcie liczenia, do magazynowania danych i wyników częściowych. Oczywiście pamięć



Ryc. 6

jest urządzeniem elektronowym, pozwalającym na zapis i odczyt danych z szybkością od kilkuset do kilkuset tysięcy danych na sekundę.

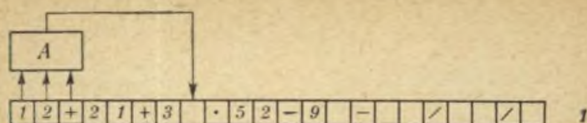
Dla naszych rozważań przyjmiemy, że pamięć to taśma papierowa, podzielona na kratki. W każdej kratce może być zapisana litera, liczba, bądź też znak działania.

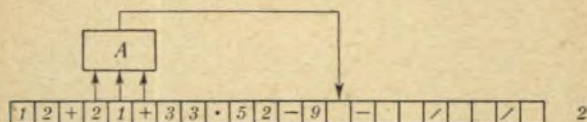
Drugą częścią maszyny jest a r y t m o m e t r, wykonujący działania arytmetyczne. Jest to również urządzenie elektronowe. W bardzo szybkich maszynach arytmometr może wykonać nawet do kilkuset tysięcy działań na sekundę.

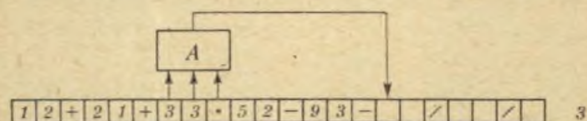
Trzecim zasadniczym elementem maszyn programowanych jest s t e r o w a n i e. Jest to urządzenie, które krok po kroku analizuje formułę, nastawia arytmometr na odpowiednie działanie, wprowadza oba argumenty do arytmometru i po wykonaniu działania umieszcza wynik na właściwym miejscu.

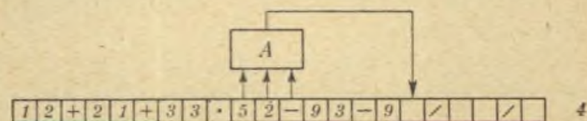
Schemat maszyny pokazany jest na rys. 6. W pamięci zapisana jest formuła. Arytmometr nastawiony jest na pierwsze działanie. Sterowania dla przejrzystości na rysunku nie pokazano. Możemy sobie wyobrazić, że sterowanie działa jak następuje:

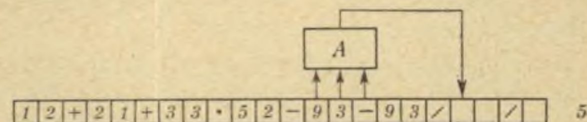
W pierwszym kroku obliczenia nastawia strzałki a_1, a_2, D na pierwsze trzy symbole zapisane na

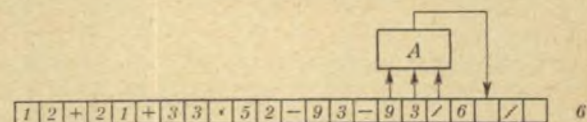

A
1 2 + 2 1 + 3 · 5 2 − 9 − / /
1


A
1 2 + 2 1 + 3 3 · 5 2 − 9 − / /
2


A
1 2 + 2 1 + 3 3 · 5 2 − 9 3 − / /
3


A
1 2 + 2 1 + 3 3 · 5 2 − 9 3 − 9 / /
4


A
1 2 + 2 1 + 3 3 · 5 2 − 9 3 − 9 3 / /
5


A
1 2 + 2 1 + 3 3 · 5 2 − 9 3 − 9 3 / 6 /
6

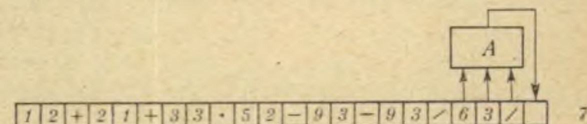

A
1 2 + 2 1 + 3 3 · 5 2 − 9 3 − 9 3 / 6 3 /
7

Рис. 7

taśmie. Będą to oba argumenty oraz pierwsze działanie. Strzałka *w* (wynik) umieszczona jest na najbliższym pustym miejscu. Sterowanie uruchamia arytmometr i po otrzymaniu wyniku zapisuje go w miejscu wskazanym przez strzałkę. W ten sposób pierwszy krok obliczenia został zakończony.

W drugim kroku sterowanie przesuwa strzałki a_1, a_2, D o trzy pozycje na prawo, odnajdując następne działanie i jego argumenty, oraz przesuwa strzałkę *w* na nowe najbliższe puste miejsce z prawej strony i reszta czynności powtarza się jak poprzednio. Działanie to jest powtarzane tak długo, aż zostanie przeanalizowana cała formuła. Na ostatnim miejscu mamy wtedy wynik końcowy.

W pracy maszyny występuje ciągle powtarzający się cykl czynności:

1. Nastawianie strzałek.
2. Wykonanie obliczenia przez arytmometr.
3. Zapisanie wyniku.

Jest to tzw. **cykl pracy maszyny.** Sterowanie rządzi więc cyklem pracy maszyny. Kolejne kroki obliczenia przedstawione są na rys. 7.

Rozważyliśmy język z porządkiem liczenia *P*. Podobnie działałaby maszyna, gdybyśmy przyjęli kolejność liczenia *W*, z tą jedynie różnicą, że szukanie miejsca, gdzie należałoby umieścić wynik byłoby nieco bardziej skomplikowane aniżeli przy kolejności *P*. Zwolennicy krzyżówek z łatwością rozwiążą ten problem.

Maszyna realizująca język uproszczony z porządkiem P

Obecnie rozpatrzymy maszynę, która składa się również z trzech podstawowych elementów: pamięci, arytmometru i sterowania. Pamięć składa się z trzech niezależnych części: pamięci danych —

D, pamięci formuły — *F* oraz pamięci wyników częściowych — *W*. Schemat maszyny pokazany jest na rys. 8. Cykl pracy maszyny jest identyczny jak w maszynie poprzedniej, tzn. składa się z pobrania argumentów, wykonania działania i umieszczenia w odpowiednim miejscu wyniku działania. Zachodzą tu jednak pewne różnice. Formuła jest zapisana w pamięci formuł *F*, dane są umieszczone kolejno w pamięci danych *D*, a wyniki częściowe są wpisywane do kolejnych kratek pamięci *W*. Działanie sterowania jest nieco bardziej skomplikowane niż poprzednio. Cykl pracy wygląda następująco:

1. Sterowanie analizuje symbol wykonywanego działania oraz symbole obu argumentów.

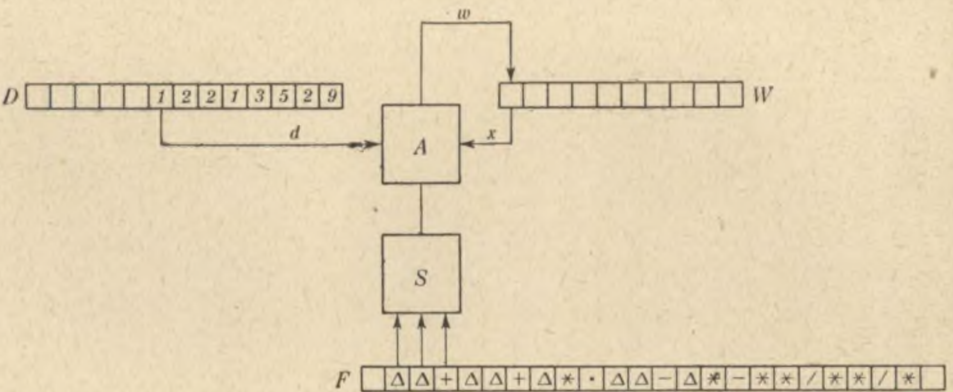
2. Arytmometr jest nastawiany na odczytane działanie oraz pobierane są do arytmometru oba argumenty. Jeżeli symbol argumentu wskazuje, że argument jest daną początkową, jest on pobierany z pamięci danych *D*; jeżeli symbol argumentu jest gwiazdką *, argument jest pobierany z pamięci wyników częściowych *W*.

3. Arytmometr wykonuje nastawione działanie.

4. Wynik działania jest zapisany w pamięci wyników częściowych *W*.

Rozpatrzmy dokładniej drugi krok cyklu, tj. pobieranie argumentów i umieszczanie wyniku częściowego. Przyjmiemy zasadę, że strzałki w pamięci *D* oraz *W* wskazują, skąd pobrać argumenty i gdzie umieścić wynik. Jeżeli z jakiejś kratki pamięci wychodzi strzałka, powiemy, że to miejsce jest przygotowane do odczytu i podobnie, jeżeli strzałka wchodzi do pamięci, to miejsce to jest przygotowane do zapisu.

Teraz możemy rozpatrzeć dokładnie proces sterowania. Istnieją cztery możliwe kombinacje argumentów Δ i *, a mianowicie: $\Delta\Delta$, Δ *, * Δ oraz **. Rozważmy działanie sterowania w każdym z tych przypadków.



Rys. 3

1. ΔΔ — oznacza, że oba argumenty są danymi. Ponieważ wszystkie dane są umieszczone kolejno, oznacza to, że jako argumenty należy odczytać dwie kolejne liczby z pamięci *D*. Symbol Δ w pamięci *F* oznacza więc: „odczytaj to, co jest napisane w zaznaczonej strzałką kratce pamięci *D* i przesuń strzałkę o jedno miejsce na prawo. Wynik zapisz w miejscu pamięci *W* oznaczonym strzałką i przesuń strzałkę o jedno miejsce w prawo".

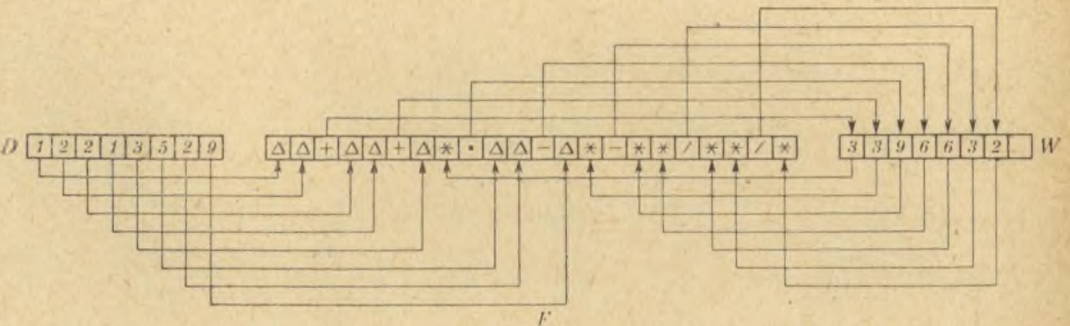
2. Δ_* — oznacza: „odczytaj jako lewy argument liczbę z miejsca pamięci *D*, zaznaczonego strzałką, i przesuń strzałkę *d* o jedno miejsce w prawo; jako drugi argument odczytaj liczbę zapisaną w miejscu zaznaczonym strzałką *x* w pamięci *W* i wynik zapisz w pamięci *W*, w miejscu zaznaczonym strzałką *w*; przesuń strzałkę o jedno miejsce w prawo".

3. *Δ — oznacza: „odczytaj jako lewy argument liczbę z pamięci *W* z miejsca oznaczonego strzałką *x*, i przesuń strzałkę *x* o jedno miejsce w prawo; jako prawy argument odczytaj liczbę z pamięci *D* z miejsca zaznaczonego strzałką *d*, przesuń strzałkę *d* o jedno miejsce w prawo. Wynik zapisz w pamięci *W* w miejscu zaznaczonym strzałką *w* i przesuń strzałkę *w* o jedno miejsce w prawo".

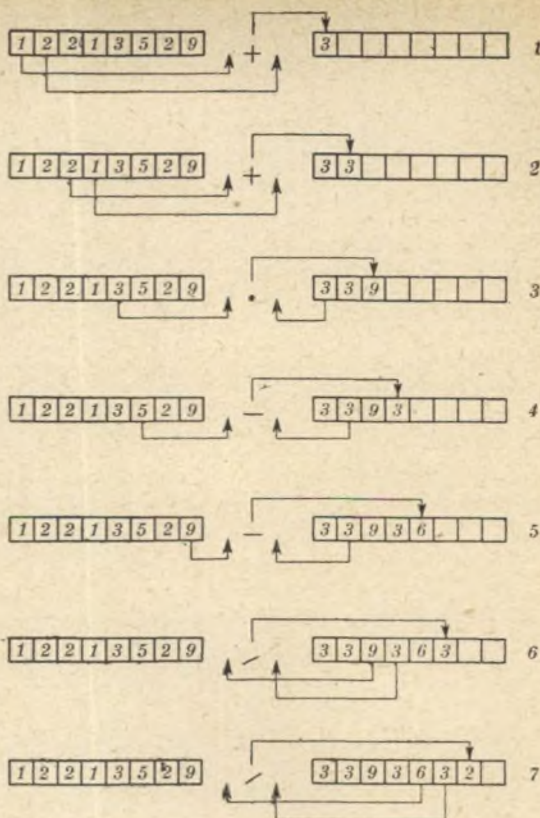
4. ** — oznacza: „odczytaj jako lewy argument liczbę z pamięci *W*, z miejsca zaznaczonego strzałką *x*, przesuń strzałkę *x* o jedno miejsce w prawo; jako prawy argument odczytaj liczbę z pamięci *W*, z miejsca oznaczonego strzałką *x*, przesuń strzałkę *x* o jedno miejsce w prawo; wynik zapisz w pamięci *W* w miejscu oznaczonym strzałką *w*, przesuń strzałkę *w* o jedno miejsce w prawo".

Mówiąc krótko: symbol Δ oznacza:

„odczytaj miejsce w pamięci *D* zaznaczone strzałką i przesuń strzałkę o jedno miejsce w prawo".



Rys. 9



Ryc. 10

Symbol * oznacza:

„odczytaj miejsce w pamięci *W* zaznaczone strzałką i przesuń strzałkę o jedno miejsce w prawo".

Wynik częściowy jest zawsze umieszczany w pamięci *W* w miejscu oznaczonym strzałką *w* i po zapisaniu strzałka *w* jest przesunięta o jedno miejsce w prawo. Znaczy to, że dane w trakcie

liczenia są pobierane kolejno z pamięci *D*; wyniki częściowe są również umieszczane kolejno w pamięci *W* oraz odczytywane w kolejności ich zapisywania.

Zamiast mówić, że strzałki są przesuwane wzdłuż taśmy, możemy również przyjąć, że po każdym zapisie i odczycie taśmy się przesuwają o jedno miejsce w lewo.

Stan pamięci *D* i *W* w kolejnych krokach obliczenia pokazany jest na rys. 9 i 10.

Maszyna realizująca język uproszczony z porządkiem W

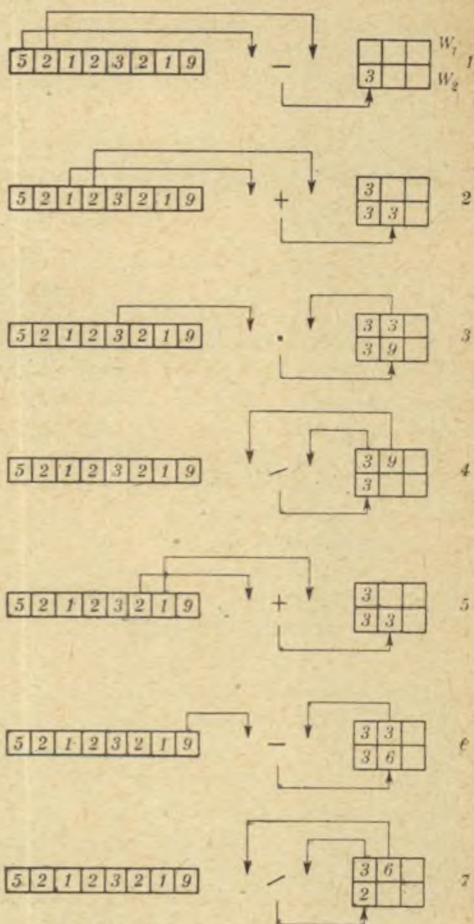
Język ten różni się od języka omawianego poprzednio tylko kolejnością wykonywania działań. Schemat maszyny jest również taki sam, jak poprzednio (rys. 8). Inny jest natomiast sposób umieszczania i pobierania wyników częściowych z pamięci *W*. Analiza przebiegu obliczenia, na podstawie drzewa da nam odpowiedź, jak należy zapisywać i odczytywać wyniki częściowe w pamięci *W*.

Zasadę umieszczania wyników częściowych można wyrazić krótko w ten sposób:

1. Jeżeli oba argumenty są danymi, to jako argumenty pobierz kolejne liczby z pamięci *D*; wynik działania zapisz na kolejne miejsce w pamięci *W*.

2. Jeżeli jeden z argumentów jest daną, a drugi wynikiem częściowym, to daną pobierz z kolejnego miejsca pamięci *D*, zaś jako argument, który jest wynikiem częściowym, pobierz ostatnio zapisaną liczbę z pamięci *W*; wynik rozpatrywanego działania zapisz w pamięci *W* na miejsce odczytanej ostatnio liczby.

3. Jeżeli oba argumenty są wynikami częściowymi, to jako argumenty pobierz dwie ostatnie



W_1 - W przed obliczeniem

W_2 - W po obliczeniu

Ryc. 11

liczby zapisane w pamięci *W* i wynik działania zapisz na miejsce liczby przedostatniej (liczby odczytywane z pamięci *W* są po odczytaniu zawsze wymazywane).

Widzimy, że wymagana pojemność pamięci wyników częściowych w maszynie wykonującej obliczenia w porządku *W*, jest mniejsza niż w maszynie liczącej w kolejności *P*, gdyż wyniki częściowe użyte do dalszego obliczenia są z pamięci wymazywane.

Warto zwrócić uwagę na jeszcze jedną różnicę między pamięciami wyników częściowych w obu maszynach. Przy kolejności wykonywania działań *P*, wyniki częściowe są odczytywane z pamięci *W* w kolejności ich zapisania, tj. liczby najpierw zapisane są również najpierw odczytywane, natomiast przy kolejności działań *W* kolejność odczytu jest odwrotna do kolejności zapisu; odczytuje zawsze ostatnio zapisany wynik częściowy.

Stany pamięci *D* oraz *W* w kolejnych krokach obliczenia pokazanego na rys. 1 przedstawia rys. 11.

Na zakończenie streśćmy krótko dotychczasowy bieg myśli. Opracowanie ogólnego schematu maszyny (organizacji maszyny) składało się z trzech etapów:

a) Dokładnego określenia obliczeń, które maszyna ma wykonywać.

b) Określenie języka służącego do opisu obliczeń.

c) Opracowania schematu maszyny interpretującej określony język.

Wykonywanie bardziej skomplikowanych obliczeń

Określając obliczenie w pierwszym paragrafie przyjęliśmy, że jest ono wykonywane jednokrotnie. Często jednak ta sama formuła jest liczona

wiele razy dla różnych wartości argumentów. Obliczenie pojedyncze można by porównać do produkcji jednostkowej, obliczenie wielokrotne do produkcji seryjnej. A więc raz rozpatrywaliśmy proces wyprodukowania jednego przedmiotu, w drugim przypadku mielibyśmy do czynienia z procesem, w wyniku którego powstaje nie jeden a wiele przedmiotów, budowanych według tego samego schematu.

Gdybyśmy chcieli zbudować język opisujący proces wielokrotnego obliczania, byłby on różny od języków dotychczas rozpatrywanych. Nie powinno to dziwić, gdyż charakter obu tych procesów jest inny. Językami takimi nie będziemy się jednak zajmować.

Niektóre prostsze przypadki „produkcji seryjnej" można jednakże opisać językami, o których była mowa w poprzednich paragrafach. Tej sprawie poświęcimy nieco uwagi.

Rozważymy krótko taki sposób wielokrotnego obliczania tej samej formuły, w którym na przemian odbywa się: wykonanie obliczenia dla zadanych wartości danych początkowych, podstawienie nowych wartości na dane początkowe, wykonanie obliczenia itd. Jak wygląda przebieg procesu liczenia w maszynie wiemy z poprzednich paragrafów.

Rozpatrzymy tylko działanie maszyny z oddzielną pamięcią wyników częściowych. Wszystkie dane, dla których ma być obliczona formuła są umieszczone kolejno w pamięci *D*.

Po każdorazowym obliczeniu formuły wymazywane są wszystkie liczby z pamięci wyników częściowych *W*, sterowanie wraca do pierwszego działania w pamięci formuły *F*, nowe dane są pobierane z dalszych miejsc pamięci *D*.

Reasumując, wielokrotne obliczanie wymaga maszyny bardziej skomplikowanej niż obliczanie jednokrotne jakiejś formuły.

Sterowanie maszyny wykonuje naprzemian dwie czynności: realizuje proces obliczenia i proces podstawiania.

Zastanówmy się teraz, jak powinien wyglądać schemat maszyny, za pomocą której możemy automatycznie wykonywać jedno za drugim różne obliczenia. Ograniczymy się również do rozważania maszyny z oddzielną pamięcią wyników częściowych, pokazanej na rys. 8.

W tym przypadku wystarczy, aby pamięć formuł F była dostatecznie duża, celem pomieszczenia wszystkich formuł w kolejności ich wykonywania. Cykl pracy maszyny nie ulega wtedy w zasadzie zmianie, jedynie po obliczeniu każdej formuły wymazywana jest pamięć wyników częściowych.

Kolejne formuły mogą być też umieszczane w pamięci *D*, na przemian z danymi.

Cykl pracy maszyny rozpoczyna się wtedy od przepisania całej kolejnej formuły z pamięci *D* do pamięci *F* i dalej przebieg obliczenia jest identyczny z poprzednim.

Każde nowe wymaganie, odnośnie możliwości maszyny, rozbudowuje jej język oraz konstrukcję. W maszynie, która miałaby mieć jakieś praktyczne zastosowanie, należałoby wziąć znacznie więcej czynników pod uwagę niż dotychczas; spowodowałoby to dalsze rozbudowanie języka i, co za tym idzie, konstrukcji maszyny.

4

Inne języki

Języki nawiasowe

Konstruując języki przydatne do opisu obliczeń, nie wzięliśmy pod uwagę języka już istniejącego, znanego wszystkim od pierwszych klas szkoły podstawowej. W języku tym obliczenie z rys. 1 zapiszemy

$$((9 - (2 + 1))/((3 \cdot (1 + 2))/(5 - 2)))$$

Wpisaliśmy tu wszystkie nawiasy, których na ogół w praktyce się nie wypisuje, jest to bowiem zbyt kłopotliwe.

Niewątpliwie nasuwa się każdemu kilka pytań: jaki jest związek języka nawiasowego z językami poprzednio określonymi? Jaki jest związek tego języka z drzewem przedstawiającym obliczenie? Czy na podstawie języka nawiasowego można również zbudować maszynę? Odpowiemy na nie zaczynając od pytania ostatniego.

Czy można zbudować maszynę pracującą w języku nawiasowym? Pozytywną odpowiedź na to pytanie dał węgierski matematyk, L. Kalmár. Opracował on maszynę, pracującą właśnie w powszechnie przyjętej symbolice nawiasowej. Nie będziemy tutaj przedstawiać koncepcji Kalmára, gdyż wymagałoby to wyjaśnienia spraw, których

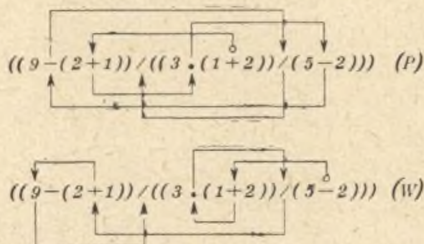
nie poruszaliśmy w niniejszej książce. Nawiążemy natomiast do zagadnień, przedstawionych w pierwszych paragrafach i opiszemy nieco inną, pracującą w języku nawiasowym maszynę.

Zgodnie z poprzednio przyjętymi zasadami pierwszą rzeczą, którą musimy ustalić przed przystąpieniem do dokładniejszej analizy języka, jest kolejność wykonywania działań. Kolejność będziemy zaznaczać, pisząc pod znakiem każdego działania odpowiednią liczbę, według drzewa przedstawiającego obliczenie. Posługując się rys. 3 i 4 działania ponumerujemy tak, jak to pokazano niżej.

/P/ ((9 − (2 + 1)) / ((3 · (1 + 2)) / (5 − 2)))
3 6 1 5 7 2 4

/W/ ((9 − (2 + 1)) / ((3 · (1 + 2)) / (5 − 2)))
2 3 1 5 6 4 7

Obliczenie rozpoczynamy zaczynając od działania o numerze największym, w naszym przypadku 7. Działanie o numerze 1 jest działaniem ostatnim. Dla łatwiejszego wzrokowego uchwycenia kolejności działań, możemy posługiwać się dodatkowo strzałkami, rysując strzałkę od każdego działania do działania po nim następującego.



Po ustaleniu kolejności działań pozostaje nam jeszcze jedna sprawa do rozstrzygnięcia: jak postępować z wynikami częściowymi? Wiemy, że przy ustalonej kolejności działań umieszczanie i odczy-

tywanie wyników częściowych nie przedstawia trudności. Musimy tylko wiedzieć, czy argumentem wykonywanego działania jest dana początkowa, czy wynik częściowy i zastosować oddzielną pamięć wyników częściowych.

Określenie, czy dany argument jest daną, czy wynikiem częściowym, w języku nawiasowym nie przedstawia trudności. Zasada jest bardzo prosta. Jeżeli z lewej strony wykonywanego działania jest liczba (lub litera), to lewy argument jest daną; jeżeli jest nawias, to argument jest wynikiem częściowym. Podobnie określamy, czy prawy argument jest daną, czy wynikiem częściowym.

Z powyższego wynika, że w formule możemy opuścić połowę nawiasów, zostawiając tylko nawiasy stojące bezpośrednio przy symbolach działań, gdyż kolejność liczenia określają nam liczby przy symbolach działań, a nawiasy mają tylko określić czy dany argument jest wynikiem częściowym. W takim uproszczonym języku nawiasowym, który nazwiemy językiem częściowo nawiasowym, formuła — pisząc działania wraz z argumentami w odstępach, dla przejrzystszego obrazu — będzie miała postać:

$$9 - (2 + 1)/(3 \cdot (1 + 2)/(5 - 2$$

Dla prostoty nie pisaliśmy numerów, oznaczających kolejność działań.

Schemat maszyny pracującej w języku nawiasowym nie różni się istotnie od schematu przedstawionego na rys. 8. Sterowanie wykonuje działania nie w tej kolejności, w jakiej są zapisane w pamięci, lecz według numerów, znajdujących się przy symbolach działań. Numery te są obliczane przez sterowanie przed rozpoczęciem obliczenia. Sposób numerowania będzie podany poniżej. Rolę symbolu * grają obecnie nawiasy. Zależnie od przyjętej kolejności obliczenia, umie-

szczanie wyników częściowych oraz ich pobieranie przebiega według recepty podanej przy opisie poprzednich maszyn. Dane są również umieszczone kolejno w pamięci danych *D*. Można również podać prostą regułę pobierania danych z pamięci *D*.

Dla wygody można do maszyny wpisywać formułę z wszystkimi nawiasami, maszyna zaś automatycznie wszystkie zbędne nawiasy opuści.

Nie trudno zauważyć związek między symboliką nawiasową oraz językiem podstawowym. Jeżeli działania wraz z argumentami wypiszemy w kolejności ich wykonywania,

(*P*) 1+2 2+1 3·(5−2 9−( ) / ( )/(

(*W*) 5−2 1+2 3·( ) / (2+1 9−( )/(

nawiasy zastąpimy symbolem * , oraz symbole działań przepiszemy na końcu każdej pary argumentów, otrzymamy formuły w języku podstawowym.

Odpowiedzieliśmy już na dwa postawione na początku pytania. Pozostało jeszcze jedno pytanie: jaki jest związek symboliki nawiasowej z drzewem przedstawiającym obliczenie? Odpowiedź na to pytanie daje rys. 12. Wyobraźmy sobie, że obchodzimy drzewo wzdłuż zakreskowanej linii, startując z punktu zaznaczonego kropką i kończąc na punkcie zaznaczonym strzałką. Po drodze obserwujemy drzewo i zapisujemy, co widzimy, według następującej reguły:

1. Jeżeli idziemy w dół gałęzi, piszemy nawias otwarty, (.

2. Jeżeli idziemy w górę gałęzi, piszemy nawias zamknięty,) .

3. Jeżeli mijamy rozgałęzienie, idąc w dół, nic nie piszemy.

4. Jeżeli mijamy rozgałęzienie idąc w górę, pi-



Ryc. 12

szemy symbol działania stojący przy mijanym rozgałęzieniu.

5. Jeżeli idziemy w górę lub w dół gałęzi końcowej (przedstawiającej daną), nawiasów nie piszemy, a piszemy tylko stojącą przy danej gałęzi liczbę lub literę.

Okazuje się, że postępując według podanych reguł napiszemy formułę w języku nawiasowym ze wszystkimi nawiasami. Jaki stąd wniosek? Język nawiasowy jest również pewnym sposobem liniowego przedstawienia struktury drzewa. Działania nie są w tym języku jednak zapisane w kolejności ich wykonywania. Nie mamy również zarezerwowanych specjalnych miejsc w formule na zapisywanie wyników częściowych, tak jak to miało miejsce w języku podstawowym.

Kolejność działań ustaliliśmy na podstawie drzewa. Można ją również ustalić bez pomocy drzewa, badając tylko rozmieszczenie nawiasów w formule. Dla przykładu podamy tylko sposób ustalenia kolejności *W*. Kolejność *W* jest jednoznacznie określona tylko lewymi nawiasami. Każdemu lewemu nawiasowi przyporządkujemy jedno działanie w następujący sposób:

1. Pierwszemu od prawej strony nawiasowi lewemu przypiszemy najbliższy z prawej strony symbol działania.

2. Każdemu następnemu lewemu nawiasowi — czytając formułę od strony prawej do lewej — przypiszemy najbliższy wolny symbol działania z prawej strony (wolny symbol — to symbol nie powiązany jeszcze z żadnym nawiasem).

Powiązanie lewych nawiasów oraz symboli działań pokazane jest poniżej.

$$((9-(2-1))/((3\cdot(1-2))/(5-2)))$$

Kolejność stawiania strzałek jest następująca:

$$((9-(2+1))/((3\cdot(1+2))/(5-2)))$$

$$((9-(2+1))/((3\cdot(1+2))/(5-2)))$$

$$((9-(2+1))/((3\cdot(1+2))/(5-2)))$$

$$((9-(2+1))/((3\cdot(1+2))/(5-2)))$$

$$((9-(2+1))/((3\cdot(1+2))/(5-2)))$$

$$((9-(2+1))/((3\cdot(1+2))/(5-2)))$$

$$((9-(2+1))/((3\cdot(1+2))/(5-2)))$$

Działania wykonujemy w kolejności odpowiadających im nawiasów, poczynając od prawej strony formuły, tj. od działania odpowiadającego pierwszemu z prawej strony nawiasowi otwartemu.

Teraz możemy już przystąpić do podania zasady numerowania działań. Najpierw ponumerujemy wszystkie nawiasy lewostronne liczbami od 1, poczynając od pierwszego nawiasu z lewej strony formuły.

```
((9−(2+1))/((3·(1+2))/(5−2)))
12  3       45 6      7
```

Chcemy, aby działania miały takie same numery, jak odpowiadające im nawiasy. Reguła jest bardzo prosta.

1. Jeżeli z lewej strony działania jest liczba (lub litera) działanie ma taki sam numer, jak poprzedzający daną liczbę nawias lewostronny.

```
Np. (9− , (2+ , (3· , (1+ , (5−
     2 2    3 3    5 5    6 6    7 7
```

2. Jeżeli z lewej strony działania znajduje się nawias (może być tylko nawias prawostronny), to numer tego działania równa się numerowi najbliższego lewostronnego nawiasu z lewej strony, zmniejszonego o ilość nawiasów prawostronnych, znajdujących się między wymienionym działaniem i nawiasem lewostronnym.

```
Np. (2+1))/ , (1+2))/
     3 3 211   6 6 544
```

Zasadę numeracji ilustruje tabelka:

```
((9−(2+1))/((3·(1+2))/(5−2)))
12 23 3 21145 56 6 5447 7
```

Jeżeli teraz dodamy do maszyny urządzenie, które będzie według podanych reguł obliczało numer każdego działania, to możemy formuły wprowadzać do maszyny w takiej formie, jak piszemy je zwykle na papierze.

Widzimy, że język nawiasowy nie jest najlepiej przystosowany do zastosowania maszynowego, przede wszystkim dlatego, że działania nie są pisane w kolejności, w jakiej mają być wykonywane. Powoduje to dwie komplikacje. Po pierwsze, musimy na podstawie formuły określić kolejność działań; po drugie — sterowanie w maszynie nie analizuje formuły kolejno działanie po działaniu, lecz szuka działań według malejących numerów. Niewątpliwie, poprzednio dyskutowane języki nie miały takich wad.

Sytuację można by częściowo poprawić, stosując wewnątrz maszyny język podstawowy, na zewnątrz natomiast — język nawiasowy. Konieczne jest wtedy w maszynie urządzenie tłumaczące z języka nawiasowego na język wewnętrzny maszyny. Każda formuła wprowadzana do maszyny przechodziłaby przez urządzenie tłumaczące i dopiero wówczas wprowadzana byłaby do pamięci maszyny. Działanie urządzenia tłumaczącego polegałoby na obliczeniu numerów działań, następnie na opuszczeniu zbędnych nawiasów, oraz uporządkowaniu wszystkich działań, wraz z odpowiadającymi im argumentami, w kolejności wykonywania.

Język wygodny dla technicznej realizacji jest niewygodny dla pisania na papierze i odwrotnie. Jaka jest tego przyczyna? Spróbujemy, częściowo przynajmniej, odpowiedzieć na to pytanie w ostatnim paragrafie.

Uproszczony język nawiasowy

Jeżeli w formule nawiasowej usuniemy wszystkie litery oznaczające liczby, to tak otrzymaną formułę nazwiemy nawiasową formułą uproszczoną.

Symbole oznaczające dane nazywamy zmiennymi. W dyskutowanych uprzednio językach

było rzeczą niezbędną wyróżnienie zmiennych po to, by wiedzieć, w które miejsca formuły należy podstawiać dane. W języku nawiasowym nawiasy grały podwójną rolę. Wyznaczały kolejność działań, oraz zaznaczały, które argumenty są wynikami częściowymi. Okazuje się, że przy pomocy nawiasów można również określić, w jakie miejsce formuły należy podstawiać dane. Tak więc, w języku nawiasowym do opisu zarówno obliczania i podstawiania wystarczą tylko nawiasy i symbole działań.

W naszym przykładzie otrzymamy w ten sposób:

$$((-(+))/((\cdot(+))/(-)))$$

Kolejność działań możemy określić tak samo, jak w symbolice nawiasowej ze zmiennymi. Odróżnienie danych od wyników częściowych określa się w sposób następujący:

Jeżeli sąsiedni nawias z lewej lub prawej strony działania jest zwrócony do działania stroną wklęsłą, to argument jest daną; jeżeli nawias jest zwrócony stroną wypukłą, to argument jest wynikiem częściowym.

Np. (—(oznacza, że lewy argument jest daną, prawy — wynikiem częściowym; (—) — oba argumenty są danymi;)—(oba argumenty są wynikami częściowymi;)—) — lewy argument jest wynikiem częściowym, prawy — daną.

Nie trudno na tej podstawie podać schemat maszyny, która będzie realizowała proces obliczenia według uproszczonej formuły nawiasowej. Umieszczanie wyników może być zrealizowane według jednego z podanych schematów, gdyż, jak wiemy, sposób umieszczania wyników częściowych w pamięci zależy tylko od porządku obliczenia, a nie od użytego języka.

Można również podać zasadę odczytywania danych na podstawie rozmieszczenia nawiasów w formule.

Zmarły niedawno polski logik, J. Łukasiewicz, wprowadził w r. 1929 sposób zapisywania formuł, który częściowo przyjął się w logice. Nie będę trudził Czytelników szczegółową analizą języka Łukasiewicza. Tym niemniej krótkie omówienie tej symboliki wydaje mi się konieczne, gdyż przeżywa ona swój renesans na gruncie maszyn matematycznych i w wielu krajach prowadzone są studia nad jej zastosowaniem. W USA zbudowano już nawet maszynę pracującą w języku Łukasiewicza; szereg innych na pewno jest w budowie.

W symbolice Łukasiewicza symbol działania piszemy nie w środku, jak to ma miejsce w symbolice nawiasowej, lecz z lewej strony argumentów. Np. zamiast a-b zapiszemy -ab. Zamiast a-$b \cdot c$ zapiszemy -$a \cdot bc$ itp.

Formuła Łukasiewicza jest także pewnym sposobem liniowego przedstawienia struktury drzewa.

Przyjmijmy, że wyniki częściowe są oznaczone symbolami odpowiadających im działań, jak to pokazano na rys. 13. Inaczej mówiąc, symbolami działań oznaczamy nie tylko rozgałęzienia, lecz również wychodzące z nich gałęzie.

Formułę Łukasiewicza można teraz otrzymać na podstawie „obchodzenia" drzewa (patrz rys. 13). Zasada jest następująca.

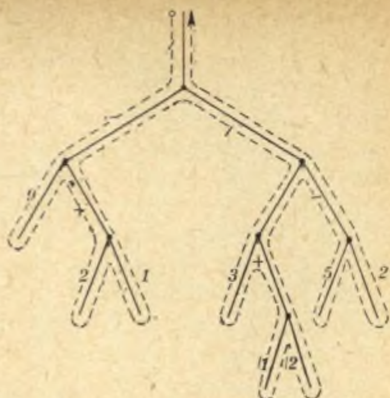
1. Jeżeli idziemy w dół gałęzi piszemy odpowiadający jej symbol.

2. Jeżeli idziemy w górę gałęzi — nic nie piszemy.

Postępując w ten sposób otrzymamy wyrażenie

$$/ -9 + 21 / \cdot 3 + 12 - 52$$

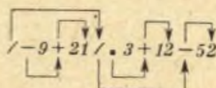
Zanim podamy sposób czytania tej formuły — kilka uwag ogólnych. W symbolice Łukasiewicza



Ryc. 13

działania zapisane są w kolejności *W* (licząc od strony prawej do lewej). Lewy argument każdego działania jest zapisany z prawej strony symbolu danego działania w ten sposób, że jeżeli symbolem lewego argumentu jest symbol działania, to argument ten jest wynikiem częściowym, jeżeli — liczba lub litera, to lewy argument jest daną. Np. / — oznacza, że lewym argumentem dzielenia jest wynik odejmowania, —9 — lewym argumentem odejmowania jest liczba 9.

Prawych argumentów działań szuka się w formule w sposób nieco bardziej skomplikowany. Dla ułatwienia posłużmy się drzewem z rys. 13 i prawy argument każdego działania zaznaczmy strzałką. Pozwoli to na wyszukanie ogólnej metody stawiania strzałek.



Kolejność pisania strzałek pokazano niżej.

/−9+21/.3+12−52

/−9+21/.3+12−52

/−9+21/.3+12−52

/−9+21/.3+12−52

/−9+21/.3+12−52

/−9+21/.3+12−52

/−9+21/.3+12−52

(Pisanie strzałek u dołu wprowadzamy tylko dla przejrzystości zapisu).

Prawym argumentem działania jest więc, jak widać, najbliższy wolny symbol z prawej strony danego działania, który nie jest symbolem lewego argumentu. Podobnie jak w symbolice nawiasowej, możemy działania ponumerować kolejnymi liczbami naturalnymi i podać prawidła znalezienia prawego argumentu na podstawie analizy formuły, bez uciekania się do pomocy drzewa.

Dalsze postępowanie z językiem jest już znane. Skoro znamy kolejność działań, wiemy jak odszukać oba argumenty i wiemy, który z argumentów jest wynikiem częściowym, możemy zastosować podany poprzednio schemat maszyny (oczywiście, w tym przypadku, z porządkiem wykonywania działań *W*).

Sterowanie maszyny będzie analizowało formułę od pierwszego działania z prawej strony, lewy agument będzie zawsze znajdował się przy wykonywanym działaniu, natomiast prawy musi być odnaleziony na podstawie numeru działania.

Z powyższego widać, że także język Łukasiewicza nie jest dla maszyn, jeśli tak można powiedzieć, „naturalny".

5

Składanie przedmiotów

Elementami rozpatrywanych do tej pory procesów były napisy, liczby, twierdzenia matematyczne. Zamiast napisów mogą występować w procesie inne dowolne przedmioty materialne, np. części jakiegoś urządzenia. Operacje będą wtedy dotyczyły montowania, a cały proces będzie polegał na złożeniu jakiegoś przedmiotu z jego części przez kolejne wykonywanie operacji. Występują wtedy te same zagadnienia, które rozważaliśmy w procesie obliczania. Można więc np. wykonywać jednocześnie kilka operacji przez różne grupy robocze, bądź też jedna grupa robocza wykonuje kolejno naznaczone operacje.

Jeżeli wiele zespołów pracuje wspólnie nad zrealizowaniem jakiegoś procesu, pojawia się problem właściwego zorganizowania pracy, podobnie jak przy wykonywaniu jednego obliczenia za pomocą kilku arytmometrów w maszynie. Sposób zorganizowania pracy w obu przypadkach w zasadzie się nie różni, obowiązują te same zasady. Jeżeli proces produkcji realizowany jest przez jedną ekipę, wyłania się problem kolejności operacji: poprzeczna czy wzdłużna, następnie — problem opisu i realizacja tego opisu. Do opisu można użyć

tych samych języków, które stosowaliśmy do opisu przebiegu obliczenia, a proces produkcji dowolnego przedmiotu możemy przedstawić za pomocą wzoru, np. w języku nawiasowym, Łukasiewicza czy innym.

Schemat, który otrzymywaliśmy z języka, będzie teraz schematem hali montażowej wskazującym, jak należy umieścić kolejno wszystkie części składowe, aby ekipa monterska, idąc wzdłuż ustawionych części i wykonując odpowiednie operacje, mogła wykonać postawione zadanie. Jeżeli zamiast ludzi operacje będzie wykonywało odpowiednie urządzenie — odpowiednik arytmometru w maszynie cyfrowej — to schematy pokazane np. na rys. 6 i 8 będą schematami maszyn, ale do automatycznej produkcji danego przedmiotu.

W dalszym ciągu rozważymy te sprawy nieco dokładniej.

Produkcja liniowa

W produkcji, zamiast mówić dane początkowe, wyniki częściowe, wyniki końcowe, mówi się: surowce, półfabrykaty, fabrykaty, albo: elementy, podzespoły, wytwory itp. Przyjmiemy terminologię ostatnią. Składanie elementów oraz podzespołów za pomocą ustalonych operacji, nazwiemy procesem produkcyjnym. Proces produkcyjny nazwiemy liniowym, jeżeli wszystkie elementy i podzespoły procesu rozłożone są w czasie procesu wzdłuż jednej linii. Proces liniowy przypomina więc liniowe przedstawienie obliczenia. Ustawienie takie ułatwia — przy pojedynczym wykonywaniu operacji — transport na właściwe miejsce podzespołów, a przede wszystkim upraszcza drogę operatora, osoby czy urządzenia, wykonującego kolejne operacje. Droga operatora jest wtedy linią prostą.

Rozpatrzmy ponownie rys. 6, traktując go teraz jako schemat produkcji pewnego wytworu. Jakie uwagi można uczynić na temat procesu przebiegającego według tego schematu? Niewątpliwie najbardziej kłopotliwą rzeczą jest tutaj transport podzespołów na właściwe miejsce. Kolejność wykonywania operacji w porządku *P*, jak i *W* jest jednakowo nieprzyjemna. Niewiele pomogłoby np. wprowadzenie dwu osób; jednej wykonującej operacje, drugiej transportującej podzespoły. Droga operatora byłaby linią prostą, natomiast transport odbywałby się ruchem wahadłowym, jak poprzednio.

Wygodniejszy byłby transport w jednym kierunku, bez powrotów (w maszynach matematycznych zasada jednokierunkowości ruchu w wielu przypadkach nie ma tak dużego znaczenia).

Czy jest to możliwe? Tak, ale tylko dla pewnego typu procesów, mianowicie dla takich procesów, w których każda operacja zawiera argument, będący elementem. Dlaczego? Wtedy bowiem każdy zespół należy przetransportować na sąsiednie miejsce.

Wyjaśnia to rys. 14. Proces ten wyraża się wzorem

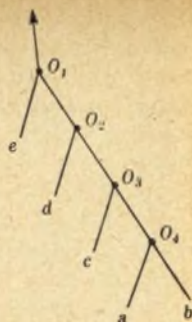
$$abO_4\, cxO_3\, dxO_2\, exO_1\, x \qquad (P)(W)$$

W procesie tym możliwa jest tylko jedna kolejność wykonywania działań. Podzespoły należy przenosić tak, jak to pokazano niżej,

$$abO_4\; cxO_3\; dxO_2\; exO_1\; x$$

Rysując formułę nieco w innej postaci,

$$\begin{matrix} ab & c & d & e \\ & x & x & x & x \\ O_4 & O_3 & O_2 & O_1 \end{matrix}$$



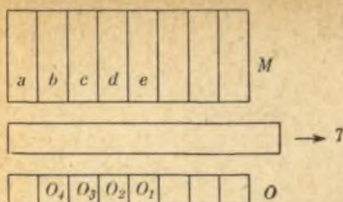
Ryc. 14

widzimy wyraźnie znaną zasadę produkcji taśmowej. Ilustruje ją rys. 15.

Magazyn *M* zawiera wszystkie potrzebne do produkcji elementy. Operator pobiera elementy *a* i *b*, wykonuje na nich operacje O_4 i otrzymany podzespół ustawia się na taśmie *T*. Taśma przesuwa się o jedną pozycję w prawo i pierwszy podzespół zatrzymuje się pod elementem *c*. Operator wykonuje operację O_3 i cykl powtarza się, jak poprzednio. Po zakończeniu wszystkich operacji wytwór znajduje się na końcu taśmy.

Widać od razu, jak można łatwo przyspieszyć produkcję. Zamiast jednego operatora można ustawić tylu operatorów, ile jest operacji. Teraz cykl pracy jest następujący. Wszystkie operacje są wykonywane jednocześnie. Następuje dostarczenie z magazynu elementów i przesunięcie taśmy produkcyjnej po czym znów następuje wykonanie operacji. Obecnie po każdym przesunięciu taśmy otrzymujemy jeden gotowy wytwór, poprzednio na jego otrzymanie potrzeba było czterech przesunięć taśmy (zakładamy tu milcząco, że czas potrzebny na wykonanie każdej operacji jest jednakowy).

Zasada ta często stosowana jest również chętnie



Ryc. 15

w maszynach matematycznych. Najwygodniej liczy się formuły o postaci jak na rys. 14.

W symbolice nawiasowej formuła taka ma postać

$$eO_1(dO_2(cO_3(aO_4b)))) \qquad 1)$$

Jak jednak postępować, jeżeli proces ma inny charakter? Każdy proces w sensie takim, jak to rozumiemy w naszych rozważaniach, da się rozłożyć na prostsze procesy, mające postać 1). Wyjaśnia to rys. 16. Proces zapisany w kolejności *P* wyrazimy formułą

*bcG deH ghI **D f*E ijF a*B **C **A *.*

Rozłożenie na procesy składowe pokazują linie przerywane na rys. 16. Otrzymamy wtedy wyrażenia

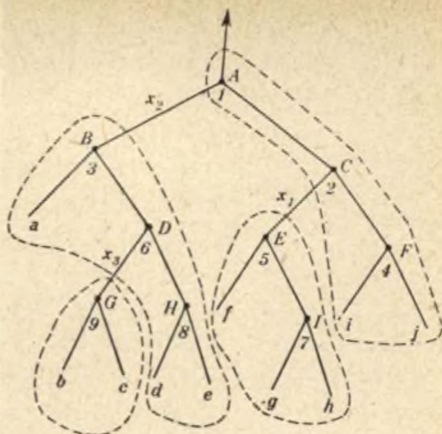
$ijF *_1*C *_2*A *$ 1

$ghI f*E *_1$ 2

$deH *_4*D a*B *_2$ 3

$bcG *_4$ 4

Podzespoły, będące wynikiem każdego procesu składowego, oznaczyliśmy gwiazdką $*$ ze wskaźnikiem. Np. rezultat procesu 4 wejdzie jako lewy argument operacji *D* w procesie 3. Każdy z procesów składowych możemy już zrealizować na zasadzie taśmowej. Argumentami operacji będą teraz



Ryc. 16

nie tylko elementy, podzespoły, ale i podzespoły powstałe w wyniku innych procesów. Podobne zagadnienie występuje w maszynach matematycznych.

Spróbujmy zapisać poszczególne procesy w symbolice nawiasowej, co pozwoli na lepsze zorientowanie się w zagadnieniu.

$$x=((aB((bGc)D(dHe)))A((fE(gIh))C(iFj)))$$

zapiszemy

$$x=(x_2A(x_1C(iFj)))$$

gdzie

$$x_1=(fE(gIh))$$

$$x_2=(aB(x_3D(dHe)))$$

$$x_3=(bGc)$$

Inną interpretacją wzoru opisującego proces produkcji taśmowej jest produkcja gniazdowa. W produkcji taśmowej elementy były nieruchome, podzespół przesuwał się na taśmie. W produkcji

gniazdowej jest odwrotnie. Podzespół jest nieruchomy, a elementy dopływają do niego. Przypomina to lepienie bałwana. Innym przykładem produkcji gniazdowej jest proces pisania. Do tekstu już napisanego dołączamy nowe znaki, otrzymując coraz to dłuższe wyrażenie.

Skręcanie powrozów

Innym przykładem produkcji jest skręcanie powrozów. Z pojedynczych włókien skręca się nitki, z nitek sznurki, ze sznurków liny, z lin powrozy. A więc drzewo może również przedstawiać proces skręcania powroza. Dalsze konsekwencje są nam już wiadome. Proces ten możemy również przedstawić w dowolnym z rozpatrywanych języków. Najbardziej wygodny wydaje się tu język nawiasowy. Różnym liczbom (czy literom) mogą np. odpowiadać nitki o różnych kolorach, działaniom — różne rodzaje splotów, np. skręcenie w prawo, w lewo czy inne.

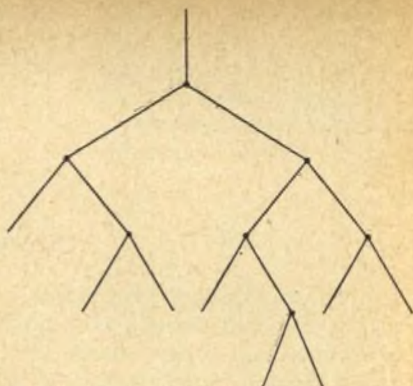
Z jednej strony mamy więc metodę zapisu produkowania nawet najbardziej skomplikowanych lin, z drugiej — nową metodę zapisu obliczania i wnioskowania.

Zamiast zapisywać przebieg obliczenia na papierze w postaci formuł, mogliby więc matematycy wyplatać powrozy.

Gdybyśmy przyjęli, że wszystkie nitki są jednakowe i że stosujemy tylko jeden sposób skręcania, np. w lewo, to język uprościł by się jeszcze bardziej. Ponieważ wszystkie działania są jednakowe, możemy ich w formule nie pisać. Oznaczmy nitkę gwiazdką. Wtedy proces skręcania zapiszemy tylko za pomocą nawiasów i gwiazdek.

Przyjmując rys. 17 jako proces skręcania sznurka, możemy napisać

((*(**))((*(**))(**)))



Ryc. 17

Zapis produkcji sznura możemy jeszcze uprościć, posługując się tylko nawiasami. Jeżeli zamiast gwiazdki będziemy pisać parę nawiasów ( ), to taki język będzie też jednoznaczny. Sznurek z rys. 17 zapiszemy wtedy

((( )(( )( )))((( )(( )( )))(( )( ))))

Pisząc zamiast nawiasu (cyfrę 1, a zamiast nawiasu) cyfrę 0, możemy też proces produkcji sznurka przedstawić za pomocą liczby

1110110100011101101000110100000

Przedstawiając tę liczbę w układzie dziesiętnym otrzymalibyśmy rekordowo krótki zapis produkcji sznura.

Jak z zapisu produkcji sznurka skonstruować maszynę powroźniczą? W zasadzie możnaby zastosować rozważane dotychczas schematy, jednakże w tym przypadku wygodniejsze są inne rozwiązania. Zostawmy ich omówienie do innej okazji.

Proces splatania sznura jest procesem syntezy. Odwrotny do niego proces rozplatania sznura — procesem analizy.

Budowa mozaiki. Składanie klocków

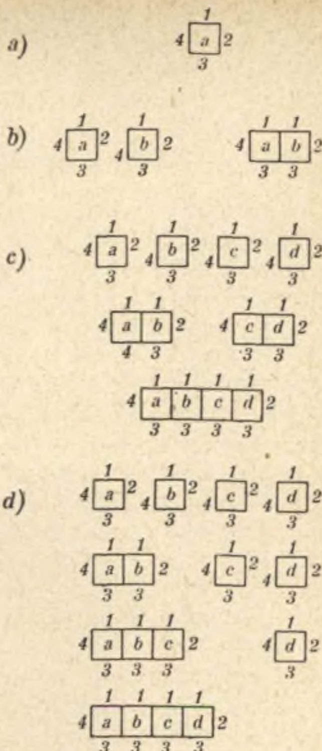
Jakkolwiek ogólna struktura rozważanych przez nas procesów jest ciągle ta sama, to jednak istnieją jakieś różnice między nimi, co wyraźnie widać przy próbie opisu procesu; nie każdy język nadaje się jednakowo dobrze do odzwierciedlenia interesujących nas własności badanego procesu. Dla wyraźniejszego podkreślenia tej zależności podamy jeszcze jeden przykład składania przedmiotów.

Spróbujmy opisać przebieg układania prostej mozaiki. Mozaika składać się będzie z płytek kwadratowych. Każdą płytkę oznaczymy małą literą, cztery krawędzie płytek ponumerujemy liczbami 1, 2, 3, 4. Krawędź górna będzie miała numer 1, krawędź prawa — numer 2, krawędź dolna — numer 3, krawędź lewa — numer 4, (patrz rys. 18a).

Krawędzie płytek nazwiemy cechami płytek. Np. cecha a_3 oznacza dolną krawędź płytki a. Wprowadzimy operację łączenia płytek krawędziami. Np. jeżeli płytki a i b połączymy ze sobą krawędziami a_2 i b_4 otrzymamy przedmiot, który będzie miał cechy a_1, b_1, b_2, b_3, a_3, a_4 (za cechy uważamy tylko krawędzie zewnętrzne, patrz rys. 18b).

Wynikiem operacji łączenia jest przedmiot, składający się w naszym przykładzie z dwu płytek. Przyjmiemy dla uproszczenia, że wynikiem operacji będziemy nazywać nie sam przedmiot, a jedną z jego cech, np. b_1, lub dowolną inną ustaloną cechę.

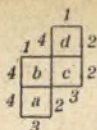
Widzimy, że tak rozumiana operacja łączenia pozwala na opisanie dowolnego procesu układania płytek. I teraz ważne są wszystkie nasze poprzed-



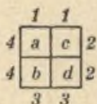
Ryc. 18

nie rozważania odnośnie języków, kolejności składania i inne. Przyjmijmy np. język podstawowy z kolejnością *P*. Wtedy przykład z rys. 18b zapiszemy: a_2b_4S. Operacje łączenia oznaczyliśmy tu literą *S*. Lewym argumentem jest przedmiot o cesze a_2, prawym argumentem — przedmiot o cesze b_4; wynikiem jest przedmiot o cechach a_1, b_1, b_2, b_3, a_3, a_4. Złożenie kafelków w jeden ciąg możemy np. opisać formułą (patrz rys. 18c)

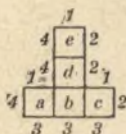
$$a_2b_4\,Sc_2d_4S\,b_2c_4S$$



$a_1\ b_3\ S\ b_2\ c_4\ S\ c_1\ d_2\ S$



$a_2\ c_4\ S\ c_3\ d_1\ S\ b_2\ d_4\ S$



$a_2\ b_4\ S\ b_2\ c_4\ S\ b_1\ d_3\ S\ d_1\ e_3\ S$

Ryc. 19

Jeżeli składanie miało przebieg jak na rys. 18d, to zapiszemy

$$a_2 b_4 S b_2 c_4 S\ c_2 d_4 S$$

Inne przykłady opisów składania kafelków pokazane są na rys. 19. Ponieważ wszystkie operacje są jednakowe, możemy symbolu operacji w formule nie pisać i wtedy dwie powyżej podane formuły będą miały postać

$$a_2 b_4, c_2 d_4, b_2 c_4$$

oraz

$$a_2 b_4, b_2 c_4, c_2 d_4$$

W identyczny sposób możemy opisać przebieg składania innych figur geometrycznych oraz tworów przestrzennych, których elementami są sześciany czy inne bryły.

Niektóre języki, np. język nawiasowy, nie nadają się do opisu tego rodzaju konstrukcji, podobnie zresztą nie można się posłużyć w tym przypadku symboliką Łukasiewicza.

Część II

Systemy

6

Systemy i hierarchie

W dotychczasowych rozdziałach rozpatrywaliśmy procesy oraz języki do ich opisu. Obecnie zajmiemy się systemami oraz językami opisującymi systemy.

Procesem, mówiąc ogólnie, nazwaliśmy zbiór przedmiotów oraz zbiór działań, wykonywanych na tych przedmiotach. **Systemem** nazwiemy zbiór przedmiotów powiązanych ze sobą jakimiś zależnościami, relacjami. Np. ojciec i syn stanowią system, gdyż między nimi zachodzi relacja pokrewieństwa. Przedsiębiorstwo jest systemem, gdyż pracownicy są podporządkowani kierownikom.

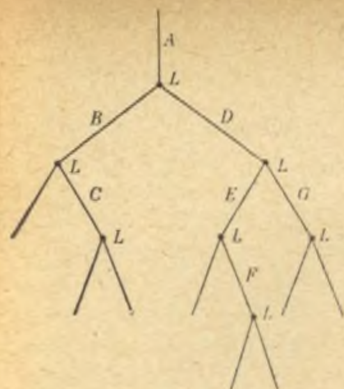
Jakkolwiek proces i system są tworami pojęciowo różnymi, to jednak zachodzi między nimi pewne formalne podobieństwo. Jedne, jak i drugie można przedstawić za pomocą odpowiednich rysunków. Zajmiemy się tylko systemami, które można przedstawić, podobnie jak obliczenia, za pomocą drzewa.

Powstaje pytanie: jak przedstawić zależności zachodzące w systemie, za pomocą języka? Ponieważ pytanie to sprowadza się do tego, jak przedstawić drzewo w języku, wszystkie rozważania dotyczące

zapisu obliczeń będą w zasadzie obowiązywały i tutaj. Przykłady wyjaśnią sprawę nieco dokładniej.

Struktura sznurka

Zamiast system będziemy też mówić **struktura.** Jak zapisać strukturę sznurka? Tak samo, jak i jego proces skręcania. Obecnie jednak inaczej interpretujemy rozgałęzienia drzewa oraz symbole im odpowiadające w formule. Rozgałęzienia teraz nie przedstawiają operacji, lecz relacje między elementami sznura, będącymi argumentami sznura i samym sznurem po wykonaniu splotu. Sznur jest już gotowy i nie interesuje nas pytanie, jak został skręcony a interesujemy się jego budową. Oczywiście na skutek skręcania dwóch elementów, powstała między nimi następująca relacja i ich wzajemny stosunek polega na tym, że są one skręcone w splot. Podobna sytuacja jest w obliczeniach. Dodając liczby 3 i 4 otrzymujemy liczbę 7. Symbol + możemy rozumieć, jako symbol operacji dodania dwu liczb, w wyniku której otrzymamy trzecią liczbę. Między liczbami 3, 4 i 7 istnieje więc pewna relacja, związek taki właśnie, że liczba 7 jest sumą 3 i 4. Relację tę również oznaczamy symbolem +, a opisujemy ją wzorem $3 + 4 = 7$. Jednakże, jeżeli spojrzymy na rysunek odpowiadający tej relacji to widzimy, że znak + możemy interpretować jako nazwę relacji, zachodzącej pomiędzy liczbami 3, 4, 7, która w formule napisanej wyżej była oznaczona dwoma symbolami +, =. Znaczenie obu symboli jest różne. Dla wyraźniejszego uchwycenia różnicy interpretacji symboli działań w języku, rozpatrzymy dokładniej przykład sznurka z rys. 17. Założyliśmy, że sznurek ten skręcany jest za pomocą tylko jednej operacji — skręcania w lewo. A więc obecnie, między każdymi dwoma skręcanymi elementami a sznu-



Ryc. 20

rem, który powstaje w wyniku tej operacji — będącym ich splotem — będzie zachodziła jedna relacja, którą oznaczymy literą *L*. Zapisując strukturę w języku nawiasowym otrzymamy

*((*L(*L*))L((*L(*L*))L(*L*)))*

Moglibyśmy — jak już mówiliśmy — strukturę sznurka zapisać bez symbolu relacji *L*, jednakże chodzi nam o zbadanie jej znaczenia w języku, dlatego symbolu relacji nie pomijamy. Z formuły tej łatwo widać jaką sznur ma strukturę.

Zastosujmy teraz do opisu symbolikę Łukasiewicza. Otrzymamy wtedy formułę

*LL*L**LL*L**L***

Formuła ta jest mniej czytelna, jednakże z rozszyfrowaniem struktury liny nie mamy trudności. Wyobraźmy sobie teraz, że ktoś dla ułatwienia rozplótł koniec sznura, jak to pokazano na rys. 20 i wszystkie fragmenty poznaczał kartkami z różnymi literami. Strukturę można by więc zapisać teraz np. stosując symbolikę Łukasiewicza tak

*AB*C**DE*F**G***

Ktoś, kto otrzymałby taką formułę, nie wiedząc jak ona powstała, miałby trudności z odtworzeniem struktury odpowiadającego jej sznurka. Litery mógłby interpretować jako różne rodzaje splotów, a nie jako nazwy sznurków składowych. W symbolice nawiasowej nie mamy tego rodzaju kłopotów. W symbolice beznawiasowej uzyskamy jednoznaczność przez zastosowanie oddzielnych symboli dla relacji, sznurków „wyjściowych" oraz sznurków wszystkich składowych, tj. stosując język podstawowy. Np. formuła

*A LBD L*C LEG L** L*F L** L*** (−*P*)

określa już jednoznacznie strukturę rozpatrywanego sznura.

Przyjęliśmy tu kolejność, którą oznaczyliśmy — *P*, tj. kolejność poprzeczną, jednakże pisaną od symbolu relacji o numerze najmniejszym, a więc odwrotnie aniżeli to uczyniliśmy przy zapisie obliczenia. Nasuwa to przypuszczenie, że nawiasy — w obliczeniu wyznaczające kolejność działań oraz które argumenty są danymi, a które wynikami częściowymi — obecnie charakteryzują relację między poszczególnymi elementami systemu. Tak właśnie jest. Bliżej tę sprawę wyjaśnimy w następnym paragrafie.

Struktura książki

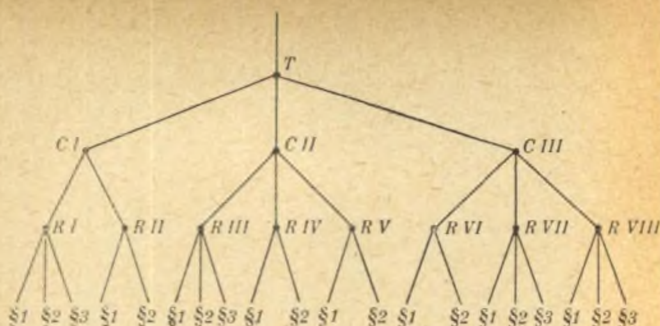
Książkę można uważać za system. Książka jest podzielona na części, części są podzielone na rozdziały, rozdziały na paragrafy itd. Struktura pewnej książki jest pokazana na rys. 21. Dla przejrzystości nie wpisano tytułów części, rozdziałów ani paragrafów a tylko ich numery. Znów mamy wiele języków do liniowego przedstawienia tej struktury. Zapiszmy ją w symbolice Łukasiewicza.

T, CI, RI, §*1*, §*2*, §*3*, *RII*, §*1*, §*2*, *CII*, *RIII*, §*1*, §*2*,

§*3*, *RIV*, §*1*, §*2*, *RV*, §*1*, §*2*, *CIII*, *RVI*, §*1*, §*2*, *RVII*, §*1*, §*2*, §*3*, *RVIII*, §*1*, §*2*, §*3*.

Na pierwszy rzut oka formuła struktury książki jest nieczytelna. Napiszmy ją jednak w inny sposób — z góry w dół.

- *T* ....
  - *CI* ....
    - *RI* ....
      - §1 ....
      - §2 ....
      - §3 ....
    - *RII* ....
      - §1 ....
      - §2 ....
  - *CII* ....
    - *RIII* ....
      - §1 ....
      - §2 ....
      - §3 ....
    - *RIV* ....
      - §1 ....
      - §2 ....
    - *RV* ....
      - §1 ....
      - §2 ....
  - *CIII* ....
    - *RVI* ....
      - §1 ....
      - §2 ....
    - *RVII* ....
      - §1 ....
      - §2 ....
      - §3 ....
    - *RVIII* ....
      - §1 ....
      - §2 ....
      - §3 ....



Ryc. 21

Zamiast kropek wpiszmy tytuły odpowiednich części, rozdziałów i paragrafów i otrzymamy spis rzeczy. Spis treści książki jest więc formułą Łukasiewicza. Warto zwrócić uwagę, że części i rozdziały są ponumerowane według kolejności poprzecznej.

W zasadzie w tekście moglibyśmy nie wypisywać ponownie tytułów podanych w spisie rzeczy, wystarczyłoby tylko, gdybyśmy każdy paragraf oddzielali od siebie jakimkolwiek znakiem. Na podstawie spisu rzeczy łatwo można by wtedy ustalić, które paragrafy stanowią wspólne rozdziały i części. Wygodne by to nie było, ale to już inna sprawa.

Można by też postąpić odwrotnie. Nie pisać spisu rzeczy, a odpowiednie tytuły paragrafów, rozdziałów i części umieścić w tekście. I teraz książkę można uważać za formułę Łukasiewicza, lecz w postaci odpowiadającej formule, w której wartości są podstawione bezpośrednio do formuły, np. $/- 9 + 21/ \cdot 3 + 12 - 52$, tj. książka miałaby wzór

T CI, RI, §1, a, §2, b, §3, c, RII, §1, d, §2, e, itd.

Litery *a*, *b*, *c*, *d*, *e* oznaczają teraz treści odpowiednich paragrafów. Celem jest tutaj wygoda czytającego. Tytuły części, rozdziałów oraz paragrafów grają więc w książce identyczną rolę jak

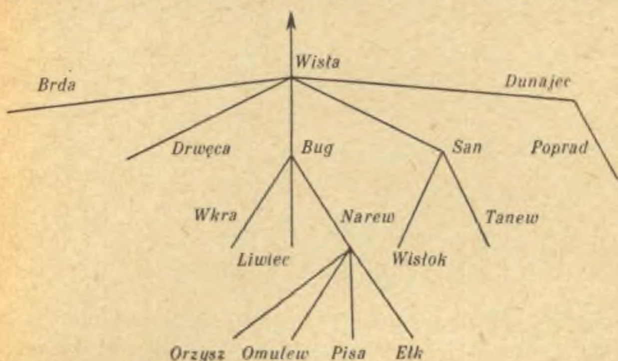
symbole działań czy relacji w formule matematycznej.

Książka jest więc również formułą, tylko w innym języku niż formuły matematyczne. Formalnie rzecz biorąc, między formułą matematyczną w języku Łukasiewicza a książką nie ma różnicy. Dobre zrozumienie tego przykładu pozwoli na uchwycenie istotnej różnicy między maszynami przedstawionymi na rys. 6 i 8. Maszyna na rys. 6 czyta jakgdyby książkę bez spisu rzeczy, natomiast maszyna na rys. 8 „czyta spis rzeczy" i na jego podstawie wynajduje właściwe fragmenty w tekście — pamięci danych.

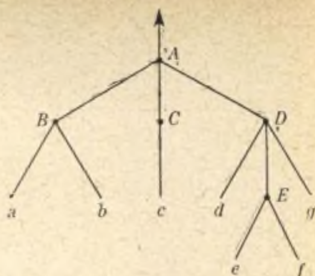
Analogie między książką a formułą Łukasiewicza łatwiej sobie wyobrazić, jeżeli przyjmiemy, że kartki nie są zszyte, a ułożone jedna obok drugiej. Wtedy już analogia nie powinna budzić wątpliwości.

Dorzecze

Ciekawym przykładem systemu jest dorzecze. Relacją wiążącą rzeki dorzecza jest „być dopływem". Rozpatrzmy dokładniej dorzecze Wisły, przedstawione na rys. 22. Dla uproszczenia nie



Ryc. 22



Ryc. 23

rozpatrujemy wszystkich rzek dorzecza Wisły, a tylko ważniejsze. Liniowe przedstawienie dorzecza jest także formułą w dowolnym z języków rozpatrywanych poprzednio.

Zanim przystąpimy do omówienia dorzecza, rozpatrzymy dokładniej sposób zapisywania drzew, posiadających w rozgałęzieniach różną od trzech ilość gałęzi. Przy takich drzewach, aby formułę można było odczytać jednoznacznie, musimy wiedzięć ile argumentów posiada każdy symbol relacji.

Wyjaśnia to dokładniej rysunek 23. Zapisanie tego rysunku w symbolice nawiasowej przy dotychczasowych założeniach jest oczywiście niemożliwe. W języku podstawowym drzewo to zapiszemy

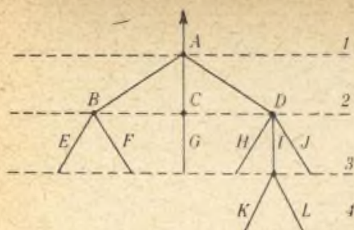
*efE abB cC a*gD ***A ** (P)

*efE d*gD cC abB ***A ** (W)

Gdybyśmy chcieli zastosować język np. Łukasiewicza, konieczne jest przy każdym symbolu relacji podanie ilości argumentów. A więc drzewo będzie miało postać

$$A_3B_2abC_1cD_3dE_2efg$$

Łatwo sprawdzić, że bez tych dodatkowych informacji jednoznaczne odczytanie formuły jest niemożliwe.



Ryc. 24

Podobnie jak to robiliśmy przy opisywaniu struktury sznurka — *A*, *B*, *C*, *D*, *E* możemy uważać za nazwy gałęzi a nie relacji. Wtedy konieczne jest wprowadzenie dodatkowego symbolu, oznaczającego relacje „bycia dopływem", np. literą *S*. Wtedy otrzymamy

1) $$A_3SB_2SabC_1ScD_3SdE_2Sefg$$

W przypadku porządku wzdłużnego jeszcze inny język okaże się wygodny. Zasada jego jest przedstawiona na rys. 24.

Wszystkie gałęzie są oznaczone dużymi literami. Drzewo jest podzielone na piętra. Każde piętro — licząc od góry — jest oznaczone numerem. Łatwo sprawdzić, że jeżeli wypiszemy wszystkie litery w kolejności *W*, oraz przy każdym symbolu zaznaczymy jego piętro, to tak otrzymana formuła również jednoznacznie daje się odczytać, tj. jednoznacznie opisuje strukturę drzewa. Formuła będzie więc miała teraz postać:

2) $$A_1B_2E_3F_3C_2G_3D_2C_2H_3I_3J_3K_4L_4$$

Zastosujemy teraz formuły 1) i 2) do opisu dorzecza Wisły. Poniżej podano opis za pomocą formuły 1). Liczby z lewej strony oznaczają ilość dopływów rzeki, odpowiadają więc wskaźnikom we wzorze 1).

5 Wisła
0 Brda
0 Drwęca
3 Bug
0 Wkra
0 Liwiec
4 Narew
0 Orzysz
0 Omulew
0 Pisa
0 Ełk
2 San
0 Wisłok
0 Tanew
1 Dunajec
0 Poprad

Odszyfrowanie struktury dorzecza z formuły nie przedstawia trudności.

Jeżeli zastosujemy wzór 2) przedstawienie dorzecza jest jeszcze prostsze. Po prawej stronie zamiast wskaźników porobiono odstępy od marginesu, o wielkości równej wskaźnikowi. Jest również dokładniejsze wyjaśnienie zapisu spisu rzeczy w książce.

1 Wisła	Wisła
2 Brda	Brda
2 Drwęca	Drwęca
2 Bug	Bug
3 Wkra	Wkra
3 Liwiec	Liwiec
3 Narew	Narew
4 Orzysz	Orzysz
4 Omulew	Omulew
4 Pisa	Pisa
4 Ełk	Ełk
2 San	San
3 Wisłok	Wisłok
3 Tanew	Tanew
2 Dunajec	Dunajec
3 Poprad	Poprad

W podobny sposób możemy przedstawić strukturę przedsiębiorstwa, różne systemy klasyfikacji i inne. Co więcej, badanie niektórych własności systemów możemy sprowadzić do badania właławłasności formuł w odpowiednim języku.

Strukturę dowolnego przedmiotu, który składa się części *A*, *B*, *C*..., te zaś z kolei z dalszych części itd. również można przedstawić identycznie. Wyjaśnijmy to bliżej. Wyobraźmy sobie, że chcemy opisać budowę samochodu. Przez budowę rozumiem tu nie proces budowania, lecz strukturę. Cały samochód możemy narysować w postaci drzewa, wykazując zależności między jego częściami. Możemy zastosować to samo rozumowanie, które przeprowadziliśmy np. przy opisie dorzecza i wypisać wszystkie części samochodu według jednego z podanych wzorów. Do każdej części może być dołączony opis jej własności czy działania. Całość tworzy wtedy instrukcję, czy też opis budowy samochodu. Taki podręcznik można uważać za formułę samochodu. Ogólniej, każdy opis systemu według podanych schematów można uważać za formułę tego systemu. W opisie takim zawsze występuje problem kolejności *P* czy *W*. Czy najpierw opisywać sprawy ogólne, potem bardziej szczegółowe itd. — mamy wtedy kolejność poprzeczną, czy też najpierw opisać jedną sprawę do samego końca, następnie przystąpić do opisu kolejnej sprawy i opisać ją do końca itd. Mamy wtedy kolejność wzdłużną. Pisząc referat, wygłaszając przemówienie, opisując jakąś sytuację zawsze spotykamy się z tym problemem. Opisując krajobraz możemy np. mówić: widzimy rzekę, górę i niebo. Na rzece płyną barki, na górze jest schronisko, na niebie leci samolot. W barkach jest zboże, w schronisku turyści, w samolocie pasażerowie itd. itd. Zastosowaliśmy tu kolejność poprzeczną.

Ten sam krajobraz możemy też opisywać według kolejności podłużnej: widzimy rzekę, po której płyną barki. W barkach znajduje się zboże. Dalej widzimy górę ze schroniskiem. W schronisku są turyści. Na niebie leci samolot, wiozący pasażerów. Oczywiście, tego naiwnego przykładu nie należy brać zbyt dosłownie, jednakże ogólny schemat relacji da się wyróżnić w każdym opisie. Często jest w takich przypadkach stosowana nie jedna kolejność *P* czy *W* w całym opisie, lecz w zależności od takich czy innych czynników kolejność *P* jest zmieniana na *W* i odwrotnie. *A.* Ehrenfeucht podał symbolikę matematyczną, pozwalającą obie kolejności stosować na zmianę. Taką zmienną kolejność można więc również stosować w obliczeniach, co, jak się wydaje, może być czasem korzystne.

7

Semantyczne i syntaktyczne określenie języka

Dotychczas rozpatrywaliśmy procesy, systemy oraz sposoby ich zapisywania. Zauważyliśmy, że języki procesów i języki systemów formalnie są identyczne, jednakże nie wszystkie języki są jednakowo dobre do zapisu konkretnego systemu czy procesu. Wybór właściwego języka wymaga uwzględnienia specyfiki procesu. Opis systemu zależał od rodzaju przedmiotów i relacji między nimi. Taki sposób określenia języka, w którym punktem wyjściowym jest jakiś konkretny proces, bądź system, nazwiemy **semantycznym.** Semantyka jest nauką o związku języka z pojęciami, które on opisuje.

Język można również badać niezależnie od tego, co poszczególne formuły oznaczają, a jedynie badać takie własności formuł, które zależą tylko od ich struktury. Ten dział nazywany jest **składnią** albo **syntaktyką.** W dalszym ciągu spróbujmy spojrzeć na wprowadzone do tej pory języki z obu wymienionych punktów widzenia.

Semantyka

Zacznijmy od systemu. Systemem nazwaliśmy zbiór przedmiotów oraz zachodzące między nimi re-

lacje. Na czym polega fakt odzwierciedlenia za pomocą formuły stosunków, zachodzących między przedmiotami systemu? Jeżeli w formule opisującej jakiś system poprzestawiamy symbole, to formuła taka już tego systemu nie opisuje. Może opisywać inny system, a może też nie opisywać żadnego. Łatwo to sprawdzić, próbując na podstawie „poprzestawianej" formuły rysować drzewo. A więc relacje w systemie są w jakiś sposób związane z porządkiem symboli w formule. Relacjom między przedmiotami odpowiadają relacje między symbolami. Jeżeli jakieś przedmioty są w określonej relacji w systemie, to litery oznaczające te przedmioty są odpowiednio ułożone w formule. A więc zachodzi między nimi również pewna relacja. Formuła również stanowi system. Przedmiotami tego systemu są symbole, relacjami zaś np. bycie sąsiadem (kiedy dwie litery są obok siebie), określona odległość między literami i inne podobne relacje, związane z liniowym rozmieszczeniem liter w formule. Formuła jest więc jak gdyby modelem systemu, który przedstawia. Ponieważ w systemie może występować wiele różnych relacji, są one w formule dodatkowo zaznaczane specjalnymi symbolami. Nie zmienia to jednak faktu, że główny ciężar, jeśli tak można powiedzieć, zaznaczenia — między którymi przedmiotami zachodzą relacje — przyjmuje struktura formuły. Symbole relacji są bardziej szczegółową informacją o rodzaju relacji. Jeden system zastępujemy więc innym systemem, liniowym systemem symboli. Analizując rozmieszczenie symboli w formule, tj. relacje zachodzące między nimi możemy wykazać, jakie relacje zachodzą między przedmiotami systemu.

Rozważmy teraz procesy. W wyniku każdej operacji procesu powstaje nowy przedmiot. Przedmiot ten jest w pewnej relacji do przedmiotów, z których został złożony. Relacja ta jest określona

operacją. Po zakończeniu więc procesu powstaje system przedmiotów. System ten możemy opisać tym samym językiem, co proces. Obie formuły są wtedy identyczne z tą różnicą, że symbole operacji są teraz symbolami odpowiednich relacji.

Relacjom w systemie odpowiadały relacje w formule. Podobnie jest z procesem. Jeżeli mamy przedmiot A, którego proces produkcji opisany jest formułą F_A oraz przedmiot B, którego proces produkcji opisany jest formułą F_B, oraz na przedmiotach A i B wykonujemy operację O, to proces produkcji przedmiotu C, który jest wynikiem operacji O, jest opisany formułą, która jest wynikiem pewnej operacji na formułach F_A i F_B. Np. w symbolice nawiasowej operacją tą będzie napisanie obu formuł F_A i F_B obok siebie z symbolem operacji — O pośrodku i ujęcie wszystkiego w nawias (F_AOF_B). W symbolice Łukasiewicza będzie to operacja napisania symboli A_A i F_B obok siebie z symbolem operacji O na początku OF_AF_B. W innych językach operacja na formułach jest bardziej skomplikowana i polega na napisaniu formuł F_A i F_B w ten sposób, że się one wzajemnie ze sobą przenikają. Sprawa jest prosta, ale nie będziemy jej bliżej rozpatrywali.

A więc, język procesów ma taką własność, że operacjom na przedmiotach odpowiadają operacje na formułach, tak, że z dwu przepisów otrzymujemy nowy przepis. Ponownie mamy więc pewnego rodzaju modelowanie.

Ogólnie można powiedzieć, że **formuła to model.** Języki, które rozpatrywaliśmy pozwalają więc na modelowanie procesów i systemów.

Struktura formuł

Język można również badać nie tylko z punktu widzenia znaczenia, lecz struktury. Jak wspomnie-

liśmy nie wszystkie wyrażenia mają sens. Jeżeli w formule poprzestawiamy symbole, takie wyrażenie może nie przedstawiać żadnego procesu ani systemu. Jest więc interesujące, jakie warunki musi spełniać ciąg symboli, aby był formułą **sensowną,** albo jak to się czasem mówi formułą **poprawnie zbudowaną.** Albo inaczej. Jest dane wyrażenie, jak zbadać, czy jest ono sensowne. Można zadać jeszcze inne pytanie. Np. w jaki sposób należy postępować, aby z symboli, którymi można się posługiwać w języku, zbudować formułę poprawną.

Wszystkie te pytania w zasadzie sprowadzają się do tego, jak odróżnić formuły sensowne od formuł nie mających na gruncie danego języka sensu; do badania struktury formuł sensownych, niezależnie od tego, co one oznaczają. Podamy obecnie kilka takich prostych metod badania poprawności formuł. Z pewnymi zapoznaliśmy się już w poprzednich paragrafach. Zaczniemy od najprostszej, którą można by nazwać **metodą redukcji.** Przykłady najlepiej zilustrują postępowanie redukcyjne. Każdą elementarną formułę będziemy kolejno zastępowali literą *s* (*formuła elementarna* — najmniejsza formuła poprawnie zbudowana). Jeżeli na koniec otrzymamy literę *s*, formuła jest zbudowana poprawnie, w przeciwnym przypadku — nie. W symbolice nawiasowej redukcja przebiega np. tak:

1. $(((a-b)\cdot c)/(e-d))$
2. $(((a-b)\cdot c)/s)$
3. $((s\cdot c)/s)$
4. (s/s)
5. s

Ten sam przykład w symbolice Łukasiewicza zredukujemy następująco:

1. $/.-abc-ed$
2. $/.-abcs$
3. $/\,scs$
4. $/ss$
5. s

Zastosujemy jeszcze tę metodę do języka podstawowego, z porządkiem *P*:

1. ab—*c · ed—**/*
2. sc · ed—**/*
3. ed—s*/*
4. ss/*
5. s

Podobnie postępujemy w innych językach. Ponieważ na końcu otrzymaliśmy literę *s*, formuły były zbudowane poprawnie. Zobaczmy, jak takie postępowanie wygląda w przypadku formuły niepoprawnej. Dla ustalenia uwagi zajmijmy się najpierw wyrażeniem w symbolice Łukasiewicza:

1. / · —ac—ed
2. / · —acs
3. / · ss
4. /s

Formuły 4 nie możemy zastąpić literą *s*, gdyż dzielenie jest operacją dwuargumentową, a w formule tej jest tylko jeden argument. Formuła 1 jest więc bez sensu. Podobnie sprawdzimy następującą formułę (w języku podstawowym)

1. ab—** · ed—**/*
2. s* · ed—**/*

Wyrażenia 2 nie możemy dalej upraszczać, gdyż formuła elementarna *s* * nie jest wyrażeniem sensownym. Formuła ta byłaby sensowna, gdyby zamiast gwiazdki była dowolna inna litera.

Można też zastosować metodę, polegającą na przypisaniu każdemu symbolowi formuły odpowiedniej liczby i jeżeli końcowy symbol otrzyma liczbę różną od 1, to formuła jest zbudowana niepoprawnie. Zasada ta dla symboliki nawiasowej może być np. następująca:

a) Pierwszy nawias otrzymuje liczbę 1.
b) Jeżeli przed symbolem stoi nawias (, to sym-

bol ten otrzymuje liczbę o jedną większą od nawiasu.

c) Symbol działania otrzymuje liczbę o jeden większą od poprzedzającego go symbolu.

d) Symbol po działaniu otrzymuje liczbę o jeden mniejszą od symbolu działania.

e) Nawias) otrzymuje liczbę o jeden mniejszą od poprzedzającego go symbolu.

Zastosujmy tę regułę do poprzedniego przykładu.

$$\begin{array}{c}(((a-b)\cdot c)/(e-d))\\ 1234543432323\,4\,321\end{array}$$

A więc formuła jest poprawna.
Podobne zasady można podać dla pozostałych języków. Np. dla języka podstawowego będzie ona miała postać:

a) Pierwszy symbol otrzymuje liczbę 1.

b) Litera (różna od *) otrzymuje liczbę o jeden większą od poprzedniego symbolu.

c) Symbol działania otrzymuje liczbę taką samą jak symbol poprzedni.

d) Symbol * otrzymuje liczbę o jeden mniejszą od poprzedniego symbolu.

Należy zwrócić uwagę, że ta metoda daje kryterium poprawności, jeżeli założyć, że co trzeci symbol jest symbolem działania (tzn. pozwala sprawdzać sensowność rozmieszczenia wyników częściowych).

Np: $ab-\ {*}c\cdot\ ed-\ {**}/\ {*}$

12 2 122 34 4 322 1

Metoda ta nie zależy od tego, w jakiej kolejności *P* czy *W* jest napisana formuła. A więc wyrażenia w obu językach mają identyczną strukturę.

Jak wspomnieliśmy, możemy też postępować odwrotnie. Mając zadane symbole, z jakich można budować formuły — możemy zapytać, jakie ope-

racje na symbolach prowadzą do formuł zbudowanych poprawnie. Inaczej mówiąc, chodzi tu o podanie reguł budowania formuł poprawnych. Reguły takie mają zazwyczaj tzw. postać indukcyjną. Na czym to polega wyjaśnimy na przykładach.

Zbiór symboli, z których można budować formuły, jest często nazywany alfabetem. Rozpatrzmy języki z alfabetem: *a*, *b*, ..., *x*, *y*, *z*, +,−,·,/ ,(,). Alfabet ten składa się z dwu części: liter i działań (operacji). Przecinki nie należą do alfabetu. Określimy indukcyjnie symbolikę nawiasową.

a) Pojedyncze litery są formułami zbudowanymi poprawnie.

b) Jeżeli dwie formuły zbudowane poprawnie połączymy symbolem działania i tak otrzymane wyrażenie ujmiemy w nawias, to otrzymamy formułę zbudowaną poprawnie.

Przepis ten pozwala na budowanie z symboli alfabetu formuł poprawnych. Np. (*a*-b) jest formułą poprawną, gdyż *a* i *b* zgodnie z punktem a) są formułami poprawnymi, a wyrażenie (*a*-*b*) powstało zgodnie z zasadą wymienioną w punkcie b). Na tej samej zasadzie jest formułą poprawną wzór ((*a*-b)·*a*. Postępując według punktów a) i b) możemy stwierdzić, że jest ono zbudowane poprawnie.

Indukcyjna definicja dla symboliki beznawiasowej Łukasiewicza będzie miała postać:

a) Pojedyncze litery są formułami zbudowanymi poprawnie.

b) Jeżeli dwie formuły zbudowane poprawnie napiszemy obok siebie i poprzedzimy je znakiem działania, to otrzymamy formułę zbudowaną poprawnie.

Przykład: ·*ab* jest formułą zbudowaną poprawnie, na podstawie punktu a). Podobnie -*ac*. A więc i wyrażenie /·*ab*-*ac* jest formułą zbudowaną na podstawie punktu b). Postępując podobnie można budować dalsze formuły poprawne.

Rozpatrzmy jeszcze język podstawowy. Symbol * będziemy nazywali symbolem wyróżnionym. Definicja indukcyjna będzie brzmieć:

a) Jeżeli napiszemy obok siebie dwie litery, nie będące symbolem wyróżnionym, po nich zaś napiszemy dowolny znak działania oraz symbol wyróżniony, to tak otrzymane wyrażenie jest formułą zbudowaną poprawnie.

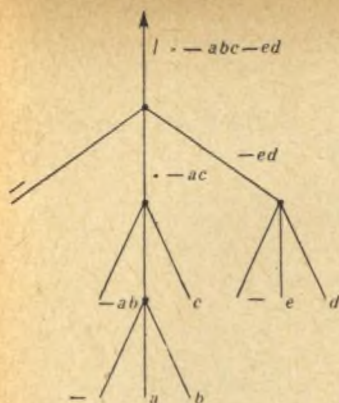
b) Jeżeli dwie formuły zbudowane poprawnie napiszemy obok siebie w ten sposób, że ostatni symbol wyróżniony pierwszej formuły napiszemy na dowolnym miejscu litery, nie będącej symbolem wyróżnionym w formule drugiej, to otrzymamy formułę zbudowaną poprawnie.

Np. *ab*-* oraz *ac*·* są formułami poprawnymi na podstawie punktu a). Wyrażenie *ab*-**c*·* jest więc formułą poprawną na podstawie punktu b). Podobnie z formuł *cd*/* oraz *ab*-**c*·* otrzymamy formułę *cd*/*ab*-**·*·

Za pomocą definicji indukcyjnych możemy budować wszystkie formuły poprawne w danym języku.

Metajęzyk

Na początku tego paragrafu powiedzieliśmy, że formuła stanowi system, gdyż między symbolami formuły zachodzą różne relacje związane z rozmieszczeniem ich w formule. Do opisu systemu możemy jednak użyć języka. Strukturę formuł możemy więc opisać w odpowiednim języku. Język taki, w którym opisujemy inny język, jest nazywany **metajęzykiem** języka opisywanego. Ponieważ budowanie formuł poprawnych w myśl definicji indukcyjnych jest — zgodnie z przyjętymi na początku określeniami — pewnym procesem, to do jego opisu możemy zbudować odpowiedni metajęzyk.



Ryc. 25

Rozpatrzmy dla przykładu język Łukasiewicza. Podamy metajęzyk, w którym będziemy mogli opisywać proces konstruowania wyrażeń poprawnych w symbolice Łukasiewicza.

W procesie konstruowania występuje jedna operacja na symbolach, polegająca na pisaniu symboli obok siebie. Oznaczymy tę operację literą *S*, od słowa stykanie. Litera *S* jest więc symbolem należącym do alfabetu metajęzyka. Jako dalsze symbole metajęzyka przyjmiemy wszyskie symbole języka Łukasiewicza, oraz gwiazdkę *.

S jest operacją trójargumentową. Pierwszym argumentem jest symbol działania, drugim i trzecim — formuła Łukasiewicza.

Proces konstruowania formuły jest przedstawiony na rys. 25. Drzewo to możemy przedstawić w dowolnym z opisywanych języków. Zastosujemy język podstawowy. Otrzymamy wtedy zależnie od kolejności dwie formuły:

$$-edS\ -abS \cdot *cS\ /**S\ * \qquad (W)$$

$$-abS\ \cdot *cS\ -adS\ /**S* \qquad (P)$$

Formuły te należy rozumieć identycznie, jak przy omawianiu procesu liczenia, tzn. wynik każdej operacji S należy umieścić na miejsce odpowiedniej gwiazdki.

Napiszemy przebieg działania dokładnie. Dla kolejności P będziemy mieli

$$-abS \cdot *cS -edS / **S *$$

$$\cdot -abcS -adS / **S *$$

$$-cdS / \cdot -abed*S *$$

$$/ \cdot -abcd-edS *$$

$$/ \cdot -abcd-ed$$

Taki sam wynik otrzymamy postępując według porządku W. Wynikiem ostatecznym jest pewna formuła w języku Łukasiewicza. Operacja S polega na przepisaniu wyrażenia stojącego przy niej na miejsce odpowiedniej gwiazdki *. Nikt w ten sposób nie będzie przepisywał symboli z miejsca na miejsce. Ale przypomnijmy sobie schematy maszyn z rys. 6 i 8. Operacje S można interpretować nie jako przepisywanie symboli na miejsce odpowiedniej gwiazdki, lecz jako dopisywanie do siebie odpowiednich fragmentów formuł, jak to ma miejsce przy definicji indukcyjnej.

Ponieważ w metajęzyku zastosowaliśmy tylko jedną operację S, możemy jej w formule nie pisać, jak to już nieraz robiliśmy uprzednio. Wtedy formuły konstruowania wyrażeń Łukasiewicza będą miały postać

$$-ed -ab \cdot *c / ** * \qquad /W/$$

$$-ab \cdot *c -cd / ** * \qquad /P/$$

Gwiazdkę można uważać za symbol operacji w metajęzyku, polegający na wpisaniu na jego miejsce symboli, tak jak to pokazano w przykładzie z symbolem operacji S. Formuły te przypominają formuły języka podstawowego, różnią się od niego tylko tym, że symbole działań umieszczone są przed, a nie po argumentach. Jeżeli symbole prze-

piszemy po prawej stronie, to otrzymamy już wyrażenia ze znanego nam języka podstawowego

*ed— ab— *c • **/ ** /W/

*ab— *c • ed— **/ ** /P/

Powyższe wzory prowadzą do napisu poprawnego w języku Łukasiewicza: + • -*abc-ed*.

Problem związku języka i metajęzyka jest ciekawy i wymaga precyzyjnych sformułowań (tak, by nie otrzymać antynomii). W tej niewielkiej książce chcieliśmy jednak tylko na przykładach zwrócić uwagę na to zagadnienie.

8

Składnia języka potocznego

Maszynowe tłumaczenie wymaga dokładnego sprecyzowania gramatyki języka, z którego tłumaczymy i języka, na który tłumaczymy oraz ich wzajemnej odpowiedniości, tj. znajomości, jakie formy gramatyczne w jednym języku odpowiadają formom drugiego języka. Nie zawsze taka ścisła odpowiedniość istnieje. Nie wszystkie reguły da się ująć w dokładne schematy. Jest to jedna z przyczyn, dla której mechaniczne tłumaczenie ma zasięg dość ograniczony.

Z punktu widzenia mechanicznego tłumaczenia ważna jest syntaktyczna analiza zdania, tj. badanie struktury zdań niezależnie od tego, co one oznaczają. Zasadniczy problem jest tutaj analogiczny do problemu występującego w językach matematycznych: jakie zestawienie słów stanowi poprawnie zbudowane zdanie; albo inaczej, jaka jest struktura zdań zbudowanych poprawnie. Treść zdań w tym przypadku nie interesuje nas zupełnie. Jako zdanie możemy również uważać wyrażenie: „Memena zakefeniła Fułę" (przykład zaczerpnięty z książki H. Greniewskiego *Elementy logiki formalnej*, Warszawa 1955, s. 117. Badanie składni zdania ma dla mechanicznego tłumaczenia zasadnicze znaczenie.

Okazuje się, że zagadnieniem tym zajmowano się już dość dawno, zresztą nie w związku z maszynami matematycznymi. Kazimierz Ajdukiewicz podał metodę badania poprawności zdań już w 1929 r.

Badanie poprawności struktury zdań można podobnie, jak badanie poprawności struktury formuł matematycznych, rozpatrywać dwojako; dany jest ciąg słów i należy zbadać czy jest on zdaniem, albo: jak ze zdań poprawnych budować dalsze zdania poprawne. Ta ostatnia metoda wymaga posiadania pewnej ilości zdań wyjściowych, uznanych już jako poprawne. Z podobną sytuacją spotkaliśmy się w indukcyjnych definicjach formuł.

Jeżeli mamy dwa zdania poprawne i połączymy je spójnikami: *i, lub, albo*, czy też innymi podobnymi, to otrzymamy zdanie poprawne.

Taka definicja zdań poprawnie zbudowanych jest prostym przykładem konstruowania zdań poprawnych ze zdań poprawnych. Jednakże z maszynowego punktu widzenia ważniejszy jest przypadek odwrotny. Mamy zadany ciąg wyrażeń, należy zbadać, czy stanowi on poprawnie zbudowane zdanie. W tym przypadku można zastosować metodę Ajdukiewicza.

Przedstawimy teraz metodę Ajdukiewicza badania poprawności zdań, nawiązując do ogólnych koncepcji przedstawionych w tej książce.

Ajdukiewicz wszystkim częściom mowy przypisuje typ logiczny. Rzeczownik ma typ logiczny oznaczony literą n (nazwa). Zdanie ma typ logiczny z (zdanie). Pozostałe części mowy mają typ logiczny ustalany w sposób następujący: jeżeli jakaś część mowy tworzy z inną częścią mowy zdanie, i druga część mowy ma typ logiczny, który oznaczymy np. przez T, to w pierwszej części mowy przypiszemy typ logiczny oznaczony ułamkiem, w którego liczniku jest litera z, a w mianowniku

— symbol oznaczający typ logiczny drugiej części mowy. Np. w zdaniu „Ala ma kota" poszczególne słowa mają następujące typy logiczne: Ala — n, ma — z/nn, kota — n. Czasownik „ma" posiada typ logiczny z/nn, gdyż z dwu nazw tworzy zdanie. Spójniki międzyzdaniowe jak *i*, *lub*, *albo* itp. mają typ z/zz, gdyż łącznie z dwoma zdaniami tworzą nowe zdanie.

W zdaniu „deszcz pada", „deszcz" ma typ n, „pada" ma — z/n, gdyż łącznie z nazwą tworzy zdanie. W zdaniu „deszcz bardzo pada" przysłówek „bardzo" ma typ

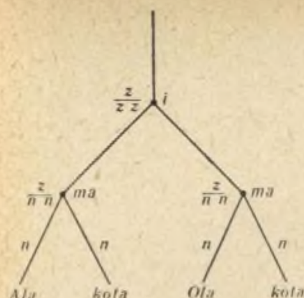
$$\frac{z}{n\,\frac{z}{n}}$$

gdyż z nazwy i z czasownika tworzy zdanie. W poprzednim zdaniu czasownik miał typ z/nn. Wiąże się to z tym, że czasownik może spełniać w zdaniu dwie funkcje: może określać relacje między dwoma przedmiotami, a może też określać stan jednego przedmiotu. Podobnie można przypisać typy logiczne innym częściom mowy.

Każdemu wyrazowi w zdaniu możemy więc przypisać typ logiczny. Zamiast badać poprawność budowy zdania, możemy na miejsce wyrazów w zdaniu wpisać ich typy logiczne i badać poprawność tak otrzymanego wyrażenia. Tutaj możemy już postępować tak samo, jak to robiliśmy w przypadku badania poprawności formuł matematycznych. Wyjaśnimy to dokładniej na przykładzie zdania „Ala ma kota i Ola ma kota". Zdaniu temu będzie odpowiadała formuła

$$n,z/nn,n,z/zz,n,z/nn,n$$

Zdanie to możemy przedstawić w postaci drzewa (patrz rys. 26). Przy każdym słowie napisany jest jego typ logiczny. Formuła zdania jest zapisana



Ryc. 26

w zasadzie w symbolice nawiasowej, jednak bez użycia nawiasów, gdyż z samej postaci zdania wiemy, jaka jest kolejność łączników. Dla ułatwienia napiszemy formułę zdania w symbolice Łukasiewicza i otrzymamy

z/zz,z/nn,n,n,z/nn,n,n

Następnie przeprowadzamy redukcję formuły zdania podobnie, jak to robiliśmy w przypadku sprawdzania formuł Łukasiewicza, chcemy bowiem sprawdzić, czy formuła ta jest poprawna. Jeżeli formuła zdania jest poprawną formułą w języku Łukasiewicza, to zdanie jest zbudowane poprawnie, w przeciwnym przypadku nie.

Redukcje przeprowadzamy w ten sposób, że wykreślamy kolejno z formuły symbole występujące w mianownikach oraz odpowiadające im symbole w reszcie formuły, jak to pokazano niżej

z/zz,z/nn,n,n,z/nn,n,n

z/zz,z/nn,n,n,z

z/zz,z,z

z

Ponieważ na końcu otrzymaliśmy literę *z*, zdanie jest zbudowane poprawnie.

Przechodzenie na symbolikę Łukasiewicza jest tutaj nieistotne, ułatwia tylko przeprowadzenie redukcji. Równie dobrze można by redukcje przeprowadzić na podstawie formuły otrzymanej bezpośrednio ze zdania.

Kryteria poprawnej budowy zdań są identyczne z kryteriami poprawnej budowy formuł matematycznych. Nic dziwnego. Skoro zdania mają opisać jakiś system, to struktura ich musi być taka sama, jak odpowiednich języków matematycznych.

Widzimy więc, że język potoczny ma taką samą strukturę, jak symbolika nawiasowa, tzn. porządek słów w zdaniu jest identyczny z porządkiem symboli w języku nawiasowym. Wyjaśnia to, dlaczego symboliki wygodne dla maszyn nie nadają się do posługiwania ręcznego. Struktura ich jest niezgodna ze składnią języka potocznego i są one dlatego trudne do zrozumienia i wymawiania. Natomiast symbolika nawiasowa jest odzwierciedleniem składni języka potocznego i dlatego posługiwanie się nią nie sprawia trudności.

Zakończenie

Spróbujemy krótko podsumować wyniki, do jakich doszliśmy. Omawialiśmy procesy obliczenia i wskazaliśmy różne możliwe sposoby ich realizacji. Pokazaliśmy, że opis tych procesów może być różnoraki. Podaliśmy szereg możliwych języków opisu. Okazało się, że od wyboru języka zależy organizacja maszyny realizującej dane obliczenie.

Dla różnych celów dogodne są różne języki — język nawiasowy dogodny przy wykonywaniu działań na papierze okazał się bardziej skomplikowany w realizacji maszynowej niż inne języki omawiane w książce.

Wskazaliśmy, że języki opisujące obliczenie nadają się do opisu innych procesów, np. procesów produkcyjnych.

Na paru przykładach omówiliśmy pojęcie systemu, sposób opisu systemu i powiązanie pojęcia systemu i procesu. Wreszcie, krótko zatrzymaliśmy się nad problemami struktury języków formalnych i naturalnych.

Wszystkie te — na pozór bardzo abstrakcyjne — rozważania mają już teraz i na pewno w daleko większym stopniu będą miały w przyszłości swoje zastosowanie praktyczne. Poznanie ogólnych zasad

dotyczących procesów, systemów i sposobów ich opisu pozwoli na zracjonalizowanie projektowania nowych, dotychczas nie istniejących maszyn czy fabryk. Pozwoli także na wyjaśnienie wielu otwartych dotąd problemów naukowych. O jednym z nich chciałbym tu powiedzieć.

Rozpatrywane systemy i procesy były bardzo proste. W przyrodzie istnieją procesy dużo bardziej skomplikowane. Np. proces budowy białka w komórce, przypominający nieco składanie przedmiotu z klocków. Przebieg tego procesu „zapisany" jest przez inny związek chemiczny, tzw. kwas DNA. Nasuwa się więc pytanie, czy struktury tego związku nie można by badać podobnie, jak badamy strukturę formuł matematycznych, czy też składni zdań w języku naturalnym?

Podobne problemy powstają w związku z opisem systemów. Czy możliwe jest liniowe opisywanie dowolnych systemów, np. struktury związków chemicznych? Wiemy, że np. H_2O jest liniowym zapisem pewnej przestrzennej kombinacji atomów wodoru i tlenu. Być może, istnieją jednak inne metody zapisu struktury związków chemicznych, bardziej przydatne niż te, które stosujemy obecnie?

Poruszana w książce tematyka nasuwa szereg problemów z najrozmaitszych dziedzin. Np. interesujące jest, na czym polega rozumienie języka? Czy może to znaczyć, że określonemu stanowi naszej świadomości odpowiadają w mózgu odpowiednie struktury chemiczne? Porozumiewanie się to jakieś liniowe przedstawienie owej struktury u nadawcy i odtworzenie na podstawie odebranej wypowiedzi identycznej struktury u odbiorcy?

Czy myślenie związane jest nieodzownie z pojęciem języka? Czy myślimy zawsze w jakimś języku, czy też myślenie jest od języka niezależne?

Zgodnie z podaną we wstępie uwagą, problematyka taka wymaga daleko idącej ostrożności, łatwo

tu bowiem popaść w rozważania, które mają niewiele wspólnego z nauką.

Poruszane zagadnienia stawiają nowe wymagania w stosunku do matematyki. Istniejące bowiem obecnie środki matematyczne nie są wystarczające do formułowania i rozwiązywania wielu problemów nie tylko w biologii, ale nawet w maszynach matematycznych. Większości problemów teoretycznych, związanych z konstrukcją maszyn matematycznych, nie udało się do tej pory rozwiązać, chociaż problemy te wydają się niezbyt trudne.

Być może omawiana problematyka spowoduje powstanie nowych gałęzi matematyki. Nie wykluczone, że maszyny matematyczne pozwolą głębiej wniknąć w istotę samej matematyki.

Tak więc, znaczenie maszyn matematycznych może być znacznie większe, niż to sobie wyobrażamy i to nie tylko dlatego, że są one doskonałym narzędziem, mającym duże znaczenie gospodarcze i naukowe, ale i dlatego, że pozwoli wniknąć w istotę wielu ważnych i dotąd nierozszyfrowanych procesów.

Indeks pojęć

algorytm 22
algorytmiczne maszyny 24
argument 9
arytmometr 25

cykl pracy maszyny 27

dana 9

formuła 22
formuła elementarna 67
formuła sensowna 66

język 22
język Łukasiewicza 42
język nawiasowy 34
język nawiasowy uproszczony 41
język podstawowy 21
język uproszczony 23

krok obliczenia 14

liniowe przedstawienie obliczeń 13

maszyna 4
maszyny algorytmiczne 29
maszyny cyfrowe 24
maszyny programowane 24
metajęzyk 21
metoda Ajdukiewicza 76
metoda redukcji 66
model 24

obliczenie 19
opis obliczeń 20
organizacja maszyn cyfrowych 24

pamięć maszyny 25
poprawność formuły 66
poprzeczny porządek 11
porządek obliczenia 10
produkcja gniazdowa 50
produkcja liniowa 47
produkcja taśmowa 48
program 2
programowanie maszyny 24

rachunek 19
redukcja 66

schemat 22
sekwencyjne obliczenia 11

semantyka 61
sensowność formuł 66
składanie przedmiotów 46
składnia 64
składnia języka potocznego 75
sterowanie 25
struktura 54
syntaktyka 64
system 54

typ logiczny części mowy 76

wzdłużny porządek obliczeń 12
wzór 22

zmienne 41

Bibliografia

K. Ajdukiewicz *Język i poznanie*. Warszawa 1960.

A. Empacher *Maszyny liczą same?* Warszawa 1960.

A. Grzegorczyk *Zagadnienia rozstrzygalności*. Warszawa 1957.

Z. Pawlak *Organizacja maszyn bezadresowych* (w druku).

B. A. Trachtenbrot *Algorytmy i automatyczne rozwiązywanie zadań*. Warszawa 1961.

„OMEGA" 1964

ukazały się:

1. L. Woolley *W poszukiwaniu przeszłości*
2. W. Michajłow *Ewolucjonizm a parazytologia*
3. D. G. Fink i D. M. Lutyens *Fizyka telewizji*
4. W. Stęślicka *Rodowód człowieka uzupełniony*
5. H. Melville *Duże cząsteczki*
6. J. Carles *Zapłodnienie*
7. I. Sachs *Drogi i manowce Świata „B"*

w przygotowaniu:

9. F. Zawielski *Czas i jego pomiary*
10. P. Lépine *Wirusy*
11. J. S. Bruner *Proces kształcenia*
12. Z. Jaworowski *Radioaktywność a zdrowie ludzkie*
13. J. Niegowski i W. Rydygier *Handel zagraniczny w gospodarce narodowej*
14. J. Rostand *Biologia twórcza*
15. J. Hadamard *Psychologia odkryć matematycznych*
16. J. Jaroszyński *Współczesne poglądy na choroby psychiczne*
17. D. Finkelsztejn *Gazy szlachetne*
18. J. Malec-Olecha *Izotopy w służbie biologii*
19. J. Chadwick *Odczytanie pisma linearnego B*
20. J. D. Williams *Strateg doskonały. Wprowadzenie do teorii gier*

Książka ta poświęcona jest niektórym problemom z zakresu maszyn cyfrowych i ogólnej teorii organizacji. Bez maszyn matematycznych nie sposób wyobrazić sobie rozwoju współczesnej nauki i nowoczesnego przemysłu; są one bowiem nie tylko niezastąpionymi narzędziami pracy, lecz również źródłem nowych idei i teorii. Budowa i programowanie maszyn matematycznych przyczyniły się do uświadomienia i sformułowania licznych problemów z zakresu ogólnej teorii struktur i organizacji. Rozwiązanie tych problemów może mieć ogromne znaczenie zarówno dla dalszych postępów w samej dziedzinie budowy maszyn, jak i dla właściwej organizacji pionów produkcyjnych i dla zrozumienia zjawisk zachodzących w żywych organizmach. Orientacja w zagadnieniach maszyn matematycznych jest więc niezbędna dla każdego, kto interesuje się wszelkiego typu strukturami o wysokim stopniu skomplikowania.

Cena w subskrypcji zł 10.—

PAŃSTWOWE WYDAWNICTWO NAUKOWE

